

VŠB – Technická univerzita Ostrava  
Fakulta elektrotechniky a informatiky  
Katedra telekomunikační techniky

Pokročilé služby v podnikové komunikaci  
Advanced Services in Enterprise  
Communication

2011

Bc. Martin Mikulec

# Prohlášení studenta

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně.  
Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

Dne:

# Poděkování

Rád bych poděkoval doc. Ing. Miroslavu Vozňákovi, Ph.D. za odbornou pomoc a konzultaci při vytváření této diplomové práce.

# Abstrakt

Tato práce se zabývá tematikou pokročilých služeb, které dnešní moderní pobočkové ústředny nabízí. Zaměřuje se především na opensource pobočkovou ústřednu Asterisk a na služby, které je v oblasti pokročilých služeb schopna nabídnout. Praktická část práce se věnuje prezenci a kalendářům ve spojení s Asteriskem. Obsahuje postup konfigurace Asterisku a ostatních služeb na virtuálním serveru a skripty pro obsluhu hovorů v závislosti na přítomnosti uživatele. Díky podrobné dokumentaci a dostupnému obrazu virtuálního stroje v příloze je možné snadno projekt rekonstruovat a navázat na něj v dalších projektech a pracích.

# Klíčová slova

Asterisk, Jabber, Kalendář, Opensource, Prezence, Pokročilé služby, SIP, Softswitch, XMPP

# Abstract

This work deals with advanced services offered by recent modern private branch exchanges (PBX). It mainly focuses on the opensource PBX Asterisk and advanced services offered by Asterisk. The practical part is aimed to presence and calendars interconnected with Asterisk. It contains configuration of Asterisk and other services on virtual server and scripts used to handling calls depending on user's presence. Thanks to detail documentation and available image of virtual machine, it is easy to reconstruct and extend this work in further projects.

# Key words

Asterisk, Jabber, Calendar, Opensource, Presence, Advanced Services, SIP, Softswitch, XMPP

# Seznam použitých symbolů, zkratek a termínů

A2Billing	Asterisk to billing - aplikace k účtování hovorů pro Asterisk
Active Directory	Proprietární systém firmy Microsoft pro centrální správu uživatelů
Agent	Operátor přiřazený do fronty obsluhy hovorů
AGI	Asterisk Gateway Interface - rozhraní pro komunikaci externích skriptů s Asteriskem
AMI	Asterisk Management Interface - rozhraní pro komunikaci s Asteriskem přes TCP/IP protokol
Apache	Nejrozšířenější HTTP server pro unixové systémy
Apache virtual host	Umožňuje hostování více domén na jednom webovém serveru
AstApi	Opensource ovladač pro operační systém Windows pro komunikaci s Asteriskem.
Brána	Rozhraní pro propojování služeb využívajících různých protokolů
CalDAV	Calendaring Extensions to WebDAV - internetový standard umožňující přístup ke kalendářům uloženým na vzdáleném serveru
Call Parking	Dočasné odložení hovoru
Callcentrum	Pokročilá služba PBX k obsluze hovorů více operátory najednou
Caller ID	Identifikace volajícího
CDR	Call Detail Record - Podrobný záznam o hovorech
Cloud	Propojení více serverů za účelem rozložení zátěže provozu
CRM	Customer relationship management - systém pro řízení vztahu se zákazníky
CSV	Comma-separated values - jednoduchý souborový formát určený pro výměnu tabulkových dat
Číslovací plán	Dialplan - řídí způsob ovládání a směrování příchozích a odchozích hovorů
Databázový server	Server pro ukládání a třídění velkého množství dat
Default	Defaultní nastavení - původní, nastavené výrobcem
DHCP	Dynamic Host Configuration Protocol - protokol k dynamickému přidělování IP adres v síti
Disc image	Obraz disku - bitová kopie spustitelná jako virtuální disk
Doplňkové služby	Služby vzniklé díky digitalizaci ústřední 4. generace
DTMF	Dual-tone multi-frequency - slouží k signalizaci přes analogové telefonní linky
Exchange Web Services	Proprietární systém firmy Microsoft pro komunikaci s Microsoft Exchange Serverem
Firewall	Síťové zařízení nebo program sloužící k řízení a zabezpečování provozu mezi sítěmi s různou úrovní důvěryhodnosti
Free BSD	Operační systém pro serverové, stolní a vestavěné počítačové platformy
FreePBX	Webové rozhraní pro jednodušší správu Asterisku
Fronta	Zajišťuje zodpovězení hovorů v pořadí, v jakém na ústřednu dorazily
FXS	Foreign eXchange Subscriber slouží jako rozhraní pro připojení analogových telefonů k Asterisku
H.264	Standard pro přenos videohovorů ve vysokém rozlišení
H.323	Protokol pro audiovizuální komunikaci

HFA	HiPath Feature Access - proprietární komunikační protokol firmy Siemens
HTTP	Hypertext Transfer Protocol - standardní protokol pro přenos dat mezi dvěma koncovými body
IAX2	Inter-Asterisk eXchange - protokol pro komunikaci mezi telefonními ústřednami Asterisk
iCalendar	Je standard (RFC 5545) pro výměnu kalendářových dat
ICQ	Proprietární protokol pro online komunikaci
IETF	Internet Engineering Task Force - komise techniky internetu vyvíjí a podporuje internetové standardy
IM	Instant Messaging - online komunikace
IP	Internet Protocol - komunikační protokol k přenosu paketů sítí
ISDN	Integrated Services Digital Network - digitální síť integrovaných služeb
IVR	Interactive Voice Response - hlasový průvodce
Java	Objektově orientovaný programovací jazyk, který vyvinula firma Sun Microsystems
JRE	Java Runtime Environment - prostředí, které je vyžadováno pro běh Javy v systému
Koncové zařízení	Obecný název pro zařízení připojené k PBX (telefon, PC, fax)
Křížový spínač	Systém k přepojování hovorů spojovacích systémů 2.generace
KVM	Kernel-based Virtual Machine - virtualizační nástroj
LCR	Least Cost Routing - směrování odchozích hovorů přes různé kanály podle cíle volání
LDAP	Lightweight Directory Access Protocol - protokol pro ukládání a přístup k datům na adresářovém serveru
LE	Large Enterprise - telefonní ústředny s počtem účastníků větším než 300
Localhost	Odkaz na právě používaný počítač
LTS	Long Term Support - dlouhodobá podpora softwaru výrobcem
Mac OS X	Operační systém firmy Apple Computer
MGCP	Media Gateway Control Protocol - komunikační protokol k propojování hovorů.
MS SQL Server	Microsoft Structured Query Language Server - databázový server firmy Microsoft
MySQL	Opensource databázový systém
NAT	Network address translation - překlad síťových adres
Nativně	Přirozeně
Open BSD	Multiplatformní otevřený operační systém
PBX	Private Branch Exchange - privátní pobočková ústředna
PDF	Portable Document Format - otevřený standard pro výměnu dokumentů
PHP	Hypertext Preprocessor - skriptovací programovací jazyk, určený především pro programování dynamických internetových stránek
PostgreSQL	Relační opensource databázový systém
Provolba	Umožňuje přímé volání na konkrétní pobočku telefonní ústředny
RAM	Random-access memory - paměť s přímým přístupem
RFC	Dokumentace internetové služby

RTP	Real-time Transport Protocol - protokol standardizující paketové doručování zvukových a obrazových dat
SCCP	Skinny Call Control Protocol - proprietární komunikační protokol firmy Cisco
Sériová linka	Služba, kdy pod jedním číslem je schováno několik klapků
SIP	Session Initiation Protocol - otevřený signalizační protokol
Skinny	Skinny Call Control Protocol - proprietární komunikační protokol firmy Cisco
SME	Small and Medium-sized Enterprise - telefonní ústředny s počtem účastníků mezi 50 - 300
SMTP	Simple Mail Transfer Protocol - internetový protokol určený pro přenos elektronické pošty
Softswitch	PBX nejnovější generace
Softwarový telefon	Desktopová aplikace jako koncové telekomunikační zařízení
SOHO	Small Office and Home Office - telefonní ústředny s počtem účastníků do 50
Spojovací pole	Část ústředny, která realizuje propojování hovorů
SSH	Secure Shell - zabezpečený komunikační protokol
Subnet	Podsít' - oddělená část počítačové sítě
TCP	Transmission Control Protocol - protokol pro spojované přenosy dat na internetu
Terminál	Příkazová řádka
Text To Speech	Aplikace pro převod textu na řeč
TLS	Transport Layer Security - protokol k zabezpečenému přenosu dat
TRUNK	Příčka mezi ústřednami
UDP	User Datagram Protocol - nespojový protokol pro přenos dat
UNISTIM	Unified Networks IP Stimulus - proprietární komunikační protokol firmy Nortel
Unix time	Čas v sekundách od půlnoci 1.ledna 1970.
URL	Uniform Resource Locator - jednoznačná adresa souboru na internetu
Virtual Adapter	Síťové rozhraní mezi virtuálním a hostujícím strojem
Virtuální stroj	Stroj, který vytváří virtualizované prostředí mezi platformou počítače a operačním systémem
VOIP	Voice over Internet Protocol - technologie umožňující přenos hlasu pomocí IP protokolu
XMPP	Extensible Messaging and Presence Protocol - protokol pro posílání zpráv a zobrazení stavu
XMPP klient	Desktopová aplikace pro online komunikaci využívající protokol XMPP
XMPP server	Serverová aplikace umožňující komunikaci pomocí protokolu XMPP

## Seznam obrázků

Obrázek 1: Operator Panel.....	6
Obrázek 2: CRM Sugar.....	7
Obrázek 3: CDR - vyhledávání záznamů.....	8
Obrázek 4: CDR - Podrobný záznam hovorů.....	8
Obrázek 5: Schéma zapojení.....	11
Obrázek 6: Propojení Asterisk a Openfire serverů.....	31
Obrázek 7: Kontrola funkčnosti propojení Asterisk a Openfire serverů.....	31
Obrázek 8: Přiřazení klapek z Asterisku XMPP uživatelům.....	32
Obrázek 9: Zaslání zprávy z Asterisku do okna XMPP klienta.....	33
Obrázek 10: Komunikace Asterisku s XMPP uživatelem.....	34
Obrázek 11: Oznámení o příchozím volání.....	36
Obrázek 12: Oznámení o nedostupnosti volaného.....	36
Obrázek 13: Schéma chování prezenčního skriptu.....	41



# Obsah

1 Úvod.....	1
2 Doplnkové služby a aplikace komunikačních systémů v korporátních sítích.....	2
2.1 Komunikační systémy v korporátních sítích.....	2
2.2 Parametry komunikačních systémů.....	3
2.3 Doplnkové služby a aplikace.....	4
2.4 IVR – hlasový průvodce.....	5
2.5 Operator Panel.....	5
2.6 CRM – Řízení vztahu se zákazníky.....	6
2.7 Účtování hovorů.....	7
2.8 Monitorování provozu a vytížení účastníků.....	7
2.9 Konferenční hovory.....	8
2.10 Propojení Asterisku s externími službami a klienty.....	9
2.10.1 Propojení s XMPP serverem.....	9
2.10.2 Propojení s kalendářem.....	9
2.10.3 Propojení s Microsoft Outlook a Windows Messenger 5.X.....	10
2.10.4 Propojení s LDAP serverem.....	10
2.10.5 Komerční řešení.....	10
3 Praktická část diplomové práce.....	11
3.1 Schéma zapojení a použité aplikace.....	11
3.1.1 Schéma zapojení.....	11
3.1.2 Wmware Workstation.....	12
3.1.3 Ubuntu 10.04.1 Server.....	12
3.1.4 Asterisk 1.8.....	13
3.1.5 Openfire 3.7.1 .....	16
3.1.6 Davical 0.9.9.....	17
3.1.7 Express Talk 4.03 a X-Lite 3.0.....	17
3.1.8 Pidgin 2.7.10 .....	18
3.1.9 Spark 2.5.8.....	18
3.1.10 Sunbird 1.0.....	18
3.2 Instalace operačního systému Ubuntu 10.04.1.....	18
3.3 Instalace softwarové ústředny Asterisk.....	20
3.3 Instalace kalendářového serveru Davical.....	22
3.4 Instalace XMPP serveru Openfire.....	25
3.5 Propojení Asterisk a Davical serveru.....	26

3.6 Připojení XMPP klientů.....	27
3.7 Připojení SIP klientů.....	28
3.8 Propojení Asterisku a Openfire serveru.....	30
3.9 Aplikace funkcí Jabberu.....	33
3.9.1 JabberSend().....	33
3.9.2 JabberReceive().....	33
3.9.3 JabberStatus() .....	35
3.10 Aplikace kalendářových funkcí.....	37
3.10.1 CALENDAR_BUSY().....	37
3.10.2 CALENDAR_WRITE().....	37
3.10.3 CALENDAR_QUERY().....	38
3.10.4 CALENDAR_EVENT().....	40
3.11 Souhrn všech funkcí – prezenční skript.....	40
4 Závěr.....	43
5 Použitá literatura.....	45
6 Seznam příloh.....	47



# 1 Úvod

Srdcem komunikace uvnitř každé větší firmy je již po několik desetiletí pobočková ústředna. Z obyčejných přepojovacích zařízení se postupem času staly moderní komunikační centra s možností tyto hlasové služby integrovat do jiných systémů. Těmto službám, které vzniknou integrací více systémů se říká služby pokročilé. Jejich užívání přináší mnoho výhod po stránce zkvalitňování komunikace mezi uživateli, přizpůsobování chodu ústředny dennímu režimu uživatelů nebo lepší obsluha zákazníků callcentra, která jsou na pokročilých službách, jako je hlasový průvodce nebo propojení s vnitřním informačním systémem, postavena.

Není to tak dlouho, co byly telekomunikace (datové a hlasové služby) proprietární produkty nebo služby kontrolované skupinou společností, které vytvářely technologie a poskytovaly tyto služby. Díky internetu a vývoji opensource programů již dnes existují rovnocenná řešení s drahými komerčními produkty, mezi které patří například Cisco Callmanager nebo Siemens HiPath ústředny. Jedním takovým opensource projektem je Asterisk. Je zdarma ke stažení a ke svému běhu vyžaduje obyčejný osobní počítač nebo server. Díky otevřenosti zdrojového kódu a obrovské komunitě vývojářů obsahuje celou řadu modulů vylepšující jeho vlastosti. Současná verze 1.8 přináší mimo jiné i podporu kalendářů, se kterými je Asterisk schopen komunikovat a řídit podle nich svůj chod. Stejně tak je schopen komunikovat s prezenčním serverem Openfire a řídit tak směrování na základě stavů uživatelů. Těmito funkcemi se zabývá praktická část práce.

V teoretické části práce se nachází rozbor některých pokročilých služeb, které je Asterisk schopen uživatelům nabídnout.

## 2 Doplnkové služby a aplikace komunikačních systémů v korporátních sítích

### 2.1 Komunikační systémy v korporátních sítích

Komunikační systém v korporátních sítích stojí převážně na centrálním bodu, kterým je pobočková telefonní ústředna, zkr. PBX (Private Branch Exchange). Ta slouží především k propojování hovorů jak mezi účastníky připojenými k PBX, tak i s účastníky ve veřejné síti. Z pohledu veřejné sítě je PBX koncovým zařízením.

Nástup podnikové komunikace se datuje k počátku 20.století a motivací k jejímu zavedení byla úspora nákladů, protože hovory uvnitř firmy jsou mezi sebou propojeny zdarma. Více literatura [4]. Z hlediska způsobu řízení a typu spojovacího pole rozlišujeme následující generace:

- 1.generace telefonních ústředen používá voliče ve spojovacím poli (např. Tesla P51)
- 2.generace používá ve spojovacím poli křížové spínače (např. Tesla PK202)
- 3.generace ústředen má centrální řízení mikropočítačem (řízení uloženým programem), spojovací pole může být řešeno různými způsoby elektronického spojování (např. Tesla UE201)
- 4.generace má buď plně anebo částečně decentralizované řízení, vždy se jedná o soustavu řídicích jednotek, které plní konkrétní úlohy, spojovací pole umožňuje změnu časové polohy. Boom v 90-tých letech 20.století. (např. Alcatel S12, Siemens Hicom 300, Avaya Definity)
- 5.generace je označována jako softswitch, pracuje s přepojováním paketů, umožňuje multimediální komunikaci a garantuje interoperabilitu s konvenčními ústřednami prostřednictvím bran. Dominantní jsou zde protokoly H323 a SIP (např. Siemens Surpass Softswitch)

Jiné dělení PBX může být také z hlediska velikosti na:

- SOHO (Small Office and Home Office) – PBX o kapacitě do 50 účastníků
- SME (Small and Medium-sized Enterprise) – kapacita 50 – 300 účastníků
- LE (Large Enterprise) – kapacita větší než 300 účastníků

Trendem doby je namísto pojmu telefonní ústředna používat pojem komunikační server. Pojem komunikační server zohledňuje skutečnost, že na dnešní spojovací systém je navázána řada aplikací, které obecně slouží ke komunikaci, ačkoliv stále zůstává stěžejním úkolem zajištění telefonního spojení.

## 2.2 Parametry komunikačních systémů

Základní kritéria, podle kterých si zákazníci svou budoucí PBX vybírají, jsou počet a typ vnitřních a vnějších linek, podporovaná koncová zařízení, škálovatelnost během růstu firmy, aplikace a doplňkové služby, náročnost obsluhy systému včetně servisních podmínek.

Počet vnitřních linek nám říká, kolik budoucích uživatelů dokáže systém obsloužit a jaká koncová zařízení je možné k PBX připojit. Mezi taková koncová zařízení můžeme řadit analogové, digitální či softwarové telefony podporující nejrůznější protokoly (H.323, SIP, HFA, IAX2, SCCP).

Počet vnějších linek (trunk) nám určuje možnosti propojení PBX s jinými ústřednami. Výhodou takového propojování je snížení nákladů na volání mimo PBX voláním přes nejvýhodnějšího poskytovatele hlasových služeb podle typu hovoru či denní doby. (LCR – Low Cost Routing)

Škálovatelnost ústředny má zásadní význam především pro začínající firmy. PBX by měla být schopna flexibilně reagovat na vzrůstající požadavky firmy co do počtu vnitřních a vnějších linek, služeb které firma požaduje a množství provozu, který PBX obsluhuje. S tím samozřejmě souvisí podpora ze strany výrobce zařízení, přehlednost administračního rozhraní PBX a kvalita servisu v případě poruchy zařízení. Nefunkčnost systému totiž může razantním způsobem negativně ovlivnit chod firmy.

Dnešní moderní PBX již neplní pouze funkci propojování hovorů, ale je schopna nabídnout mnoho doplňkových služeb již v základní výbavě. Těmito službami se zabývá následující kapitola.

## 2.3 Doplnkové služby a aplikace

S příchodem ústředn 4.generace v 90-tých letech 20.století dokázaly ústředny nabídnout celou řadu nových služeb. Více literatura [4]. Digitální systémy 4. generace umožňují nasazení ISDN, základních a primárních přípojek. Komunikace mezi ústřednami probíhá pomocí signalizace SS7. Na účastnické straně je podporována DSS1. Více literatura [1]. Mezi nejvýznamnější doplňkové služby patří:

- Automatické vyzvednutí hovoru
- Funkce nerušit
- Hlasová schránka
- Paralelní vyzvánění
- Provolba
- Přesměrování hovorů
- Převzetí hovoru
- Sériová linka
- Zobrazení nebo potlačení čísla volajícího a volaného

Tyto a další služby se za dobu jejich používání staly zcela běžné, ne-li nepostradatelné v hlasové komunikaci. Dnešní snahou ústředn páté generace je integrace služeb telefonní ústředny do jiných informačních a komunikačních systémů. Výsledkem jsou pokročilé služby, které zásadním způsobem zrychlují a zkvalitňují výměnu informací mezi uživateli. Jako praktický příklad takových služeb můžeme použít následující situaci:

Na callcentrum volá zákazník. Prostřednictvím hlasového průvodce je naveden na oddělení, se kterým požaduje problém řešit. Ústředna je propojena s CRM systémem firmy, takže podle telefonního čísla účastníka je schopna volajícího identifikovat. Zákazník má v databázi CRM systému přiřazeného soukromého agenta. Ústředna tedy zkontroluje, zdali je daný agent dostupný na telefonu, v kalendáři a XMPP klientu, a podle toho nasměruje hovor buďto přímo na něj nebo na jiného volného operátora. Operátorovi zvoní telefon a on již na displeji vidí všechny potřebné informace z CRM systému o zákazníkovi. Tyto informace mohou sloužit k autorizaci zákazníka a dalším potřebám.

## 2.4 IVR – hlasový průvodce

Mnoho lidí má spojený pojem „hlasový průvodce“ dále jen IVR (Interactive Voice Response) s automatickým průvodcem, který vás pomocí DTMF volby spojí s určitou pobočkou. Pochopitelně je tohle využití nejrozšířenější. Tato pokročilá služba však může nabídnout mnohem více než obyčejné směrování hovorů. Používá se také jako menu k přístupu k hlasovým záznamům uloženým na serveru.

Ve spojení s přístupem do databáze firmy a softwarem umožňující převod textu na hlas (Text To Speech, dále jen TTS) se jeho potenciál výrazně zvětšuje a stává se velmi silným nástrojem v přístupu k informacím. Zákazník tak může pomocí takového systému zjišťovat informace o svém účtu ve firmě, pomocí DTMF volby měnit různá nastavení a to vše automaticky bez zásahu jiné osoby do tohoto procesu, což výrazně šetří lidské prostředky. Pokud například budeme do databáze v textové podobě ukládat aktuální předpověď počasí v různých regionech, může si volající tuto předpověď nechat přečíst právě pomocí IVR systému. Na podobném principu by mohla fungovat například i zpětná vazba od zákazníka, který využil služeb callcentra. Po ukončení hovoru mu ústředna sama zavolá a požádá ho, aby pomocí DTMF volby zhodnotil kvalitu operátora, který ho obsloužil, výsledek poté není problém zapsat do databáze a následně s ním dále pracovat.

Výhodou IVR na platformě Asterisk je při nasazení ve velké společnosti také možnost tento systém jednoduše rozšířit na každou pobočku a obsluhovat tak hovory přímo na nich. Není potřeba hovory přesměrovávat do centrálního bodu, který by je obsloužil. Výpočetní prostředky jsou tak využity optimálně.

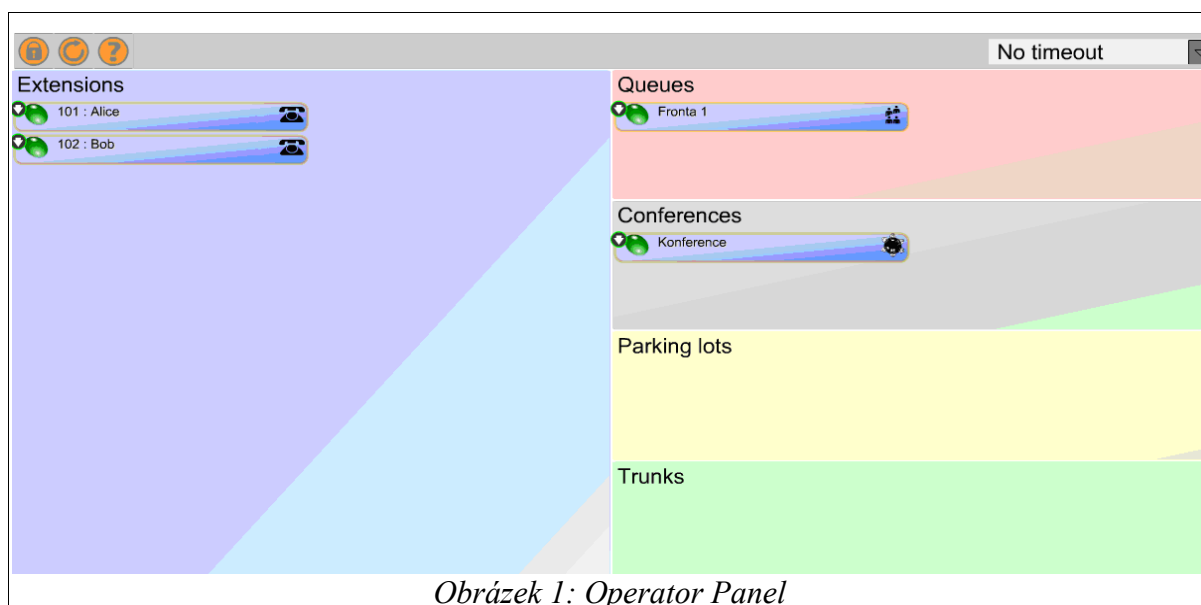
## 2.5 Operator Panel

Operator Panel je webová aplikace dostupná po instalaci nástavby FreePBX nad systém Asterisk. Podobných řešení je možno k asterisku doplnit celou řadu. Panel umožňuje zobrazit v reálném čase aktivity, které právě v ústředně probíhají. Mezi tyto informace a služby patří:

- Stavy přihlášených SIP a IAX uživatelů (dostupní, nedostupní nebo právě vyzvánějí)
- Sledování kdo s kým právě hovoří
- Naplněnost a stav konferenčních hovorů nebo fronty, do kterých jsou přihlášení jednotliví operátoři
- Ztlumení / Zesílení účastníků konferenčního hovoru
- Indikace počtu hlasových zpráv



- Přehled zaparkovaných hovorů
- Možnost zavěsit kanál či do něj vstoupit
- Přetáhnutím myši přesměrovat hovor nebo inicializovat hovor

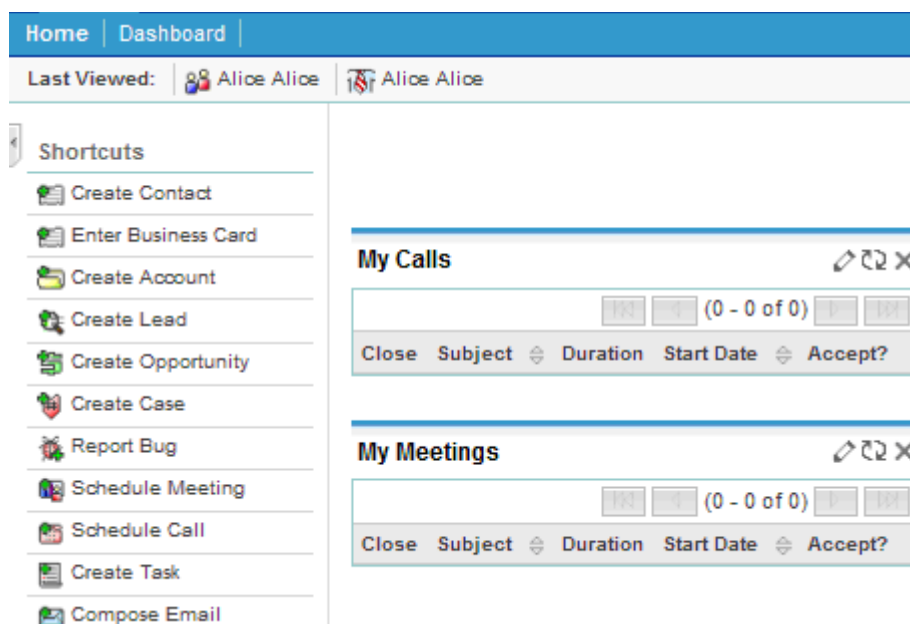


## 2.6 CRM – Řízení vztahu se zákazníky

Žádná větší společnost poskytující služby svým zákazníkům se v dnešní době neobejde bez systému řízení vztahu se zákazníky, dále CRM (Customer Relationship Management). Tento systém obsahuje veškeré informace, které má firma o svých zákaznících k dispozici a systém umožňuje s těmito informacemi přehledně pracovat. Tyto informace jsou zcela zásadní v oblastech prodeje a podpory služeb, marketingu, řízení společnosti i ekonomiky. Implementací PBX Asterisk získáváme mnoho výhod a zjednodušení komunikace jak uvnitř společnosti tak i mezi společností a jejími zákazníky. Díky propojení je možno realizovat následující služby a mnoho dalších:

- Inicializace volání přímo z webového prostředí CRM systému
- Zobrazení detailu volajícího během vyzvánění
- Směrování hovoru na konkrétního agenta na základě informací z CRM
- Monitorování historie hovorů související s volajícím číslem
- Automatické volání klientů z databáze CRM za účelem marketingu

- Zobrazení dostupnosti telefonních účastníků uvnitř CRM systému



Obrázek 2: CRM Sugar

## 2.7 Účtování hovorů

Telefonní hovory jsou účtovány již od počátku své existence. K účtování hovorů na ústředně Asterisk slouží různé nástroje, jedním z nich je opensource projekt A2Billing [13] psaný přímo pro Asterisk. Obsahuje snad všechny funkce, které jsou pro poskytovatele hlasových služeb zásadní a je ke stažení zdarma. Ke svému běhu vyžaduje kromě Asterisku, běžícím na Linux serveru, také webový server Apache a MySQL nebo PostgreSQL databázi.

## 2.8 Monitorování provozu a vytížení účastníků

Webová nástavba Asterisku FreePBX obsahuje opět díky připojení k MySQL databázi historii hovorů, dále CDR (Call Detail Record). V tomto rozhraní nalezneme informaci o přesném času volání, délce volání, kanálu, kterým byl hovor přenášen, zdrojové a cílové pobočce, jménu volajícího a stavu hovoru po ukončení (zodpovězen, selhal, atd...). Tyto záznamy je možné nejrozličnějším způsobem filtrovat a prohledávat, čímž získáme potřebné statistiky provozu jednotlivých účastníků, oddělení a kanálů za různá časová období. Stejně tak můžeme zjistit počty hovorů během určitého časového

období nebo porovnávat provoz s ohledem na celý měsíc. Jednoduše je tento nástroj dostačující pro vedoucího callcentra ke sledování vytížení oddělení a operátorů.

The screenshot shows a search interface for Call Detail Records (CDR). It includes several filter sections:
 

- Selection of the month:** From: April-2011, To: April-2011
- Selection of the day:** From: 01, To: 01, both for April-2011
- DESTINATION:** A text input field with radio buttons for Exact, Begins with, Contains, and Ends with.
- SOURCE:** A text input field with radio buttons for Exact, Begins with, Contains, and Ends with.
- CHANNEL:** A text input field.
- DURATION:** Two text input fields with radio buttons for >, equal, and <.
- Search:** A magnifying glass icon and a 'Search' button.
- Result:** Radio buttons for Minutes and Seconds.

Obrázek 3: CDR - vyhledávání záznamů

Number of calls : 5							
- Call Logs -							
	<u>Calldate</u>	<u>Channel</u>	<u>Source</u>	<u>Clid</u>	<u>Dst</u>	<u>Disposition</u>	<u>Duration</u>
1.	2011-04-19 08:32:54	SIP/102-00...	102	"Bob" <102>	102	NO ANSWER	00:09
2.	2011-04-19 08:32:51	SIP/102-00...	102	"Bob" <102>	101	FAILED	00:01
3.	2011-04-19 08:31:23	SIP/102-00...	102	"device" <102>	s	ANSWERED	00:01
4.	2011-04-19 08:30:30	SIP/102-00...	102	"Bob" <102>	1*	ANSWERED	00:14
5.	2011-04-19 08:29:46	SIP/102-00...	102	"Bob" <102>	1*	ANSWERED	00:21

Obrázek 4: CDR - Podrobný záznam hovorů

## 2.9 Konferenční hovory

Asterisk umožňuje fungovat také v roli konferenčního mostu pomocí modulu MeetMe(). Tento modul umožňuje vzájemnou komunikaci několika volajících současně jakoby byli společně v jedné místnosti. Některé z hlavních funkcí jsou:

- Vytváření konferencí chráněných heslem
  - Umožňuje spravovat probíhající konferenci – zamknout konferenci pro další účastníky
- Ovládání hlasitosti hovoru jednotlivých účastníků, odhlášení nežádoucích účastníků konference, apod.

## 2.10 Propojení Asterisku s externími službami a klienty

### 2.10.1 Propojení s XMPP serverem

Asterisk obsahuje modul jabber, který umožňuje integraci hlasové komunikace pomocí Asterisku do prostředí online komunikace prostřednictvím protokolu XMPP známý spíše pod názvem Jabber. Vzájemné propojení těchto dvou systémů přináší celou řadu výhod a nových služeb:

- Směrování hovoru na základě stavu nastaveného na XMPP klientu – lze nastavit různých uživatelských stavů, podle kterých může ústředna přizpůsobit své chování
- Asterisk může automaticky měnit stav uživatele v XMPP klientu, pokud má uživatel aktivní hovor
- Asterisk je schopen vypisovat text do okna XMPP klienta nebo číst a zpracovat odpověď, kterou do okna uživatel zadá

### 2.10.2 Propojení s kalendářem

S vývojem internetu a nástupem webových aplikací bylo možné umístit na web také kalendáře. Výhod používání kalendářů online je nepřeberné množství od rychlého vyhledávání, sdílení na webu až po synchronizaci s jinými kalendáři a aplikacemi jak na PC tak na mobilních zařízeních. Mezi nejpoužívanější protokoly pro přenos a práci s kalendáři patří CalDAV a Exchange Web Service. Právě tyto protokoly Asterisk podporuje ve svém kalendářovém modulu. Spárování Asterisku s kalendářem nám otevře následující užitečné funkce:

- Asterisk si v nastaveném časovém intervalu kontroluje dostupnost uživatele nastavenou u událostí v kalendáři. U těchto událostí je možné nastavit stav dostupný nebo nedostupný. Za základě jednoho z těchto dvou stavů přizpůsobuje směrování hovoru na daného účastníka

- Asterisk je schopen do kalendáře zapisovat. Typickým příkladem využití je záznam o probíhajících hovorech, kdy jsou do kalendáře zapsány údaje jako čas počátku a konce hovoru nebo kdo s kým v tu dobu telefonoval
- Pokud do Asterisku přidáme nějaký software pro převod textu na řeč (TTS – Text To Speech), otevrou se nám možnosti přehrávání textu zapsaného v kalendáři. Asterisk umožňuje automatické zavolání definovaný čas před začátkem schůzky poznačeným v kalendáři a přehrání textu napsaného k dané schůzce. Stejně tak si můžeme zavolat o přečtení schůzek v následující hodině

### 2.10.3 Propojení s Microsoft Outlook a Windows Messenger 5.X

K propojení Microsoft Outlook klienta s Asteriskem umožňuje AstApi – opensource Asterisk Tapi ovladač pro Windows. Tomu umožníme přístup do rozhraní pro správu Asterisku (AMI – Asterisk Managenent Interface) a díky tomu můžeme přímo z Outlooku volat kontakty v něm uložené přes ústřednu Asterisk.

V případě Windows Messenger je funkce obdobná, není však třeba používat ovladač AstApi. Více o nastavení viz literatura [5] [8].

### 2.10.4 Propojení s LDAP serverem

Asterisk je díky modulu res\_ldap schopen načíst údaje o SIP/IAX účtech a konfiguracích z LDAP databáze. Výhodou tohoto spojení je centralizovaná správa uživatelů na jednom místě v LDAP serveru. Více viz literatura [5].

### 2.10.5 Komerční řešení

Asterisk svým vývojem reaguje na trendy, které určují dominantní hráči na trhu jako je společnost Cisco Systems, Siemens a další.

Budoucnost komunikací je v tzv. sjednocené komunikaci. Jedná se o sloučení hlasových služeb, online komunikace, e-mailu, prezence, konferenčních hovorů a dalších služeb do jednoho prostředí. Tyto služby jsou dostupné na všech možných zařízeních pro zvýšení mobility uživatelů. Cisco v této oblasti nabízí produkt Cisco Unified Communication Manager v současné verzi 8.5. Společnost Siemens nabízí v tomto odvětví OpenScape UC Server. Oba systémy nabízí opravdu širokou paletu služeb. Oba tyto produkty jsou narozdíl od Asterisku zpoplatněná komerční řešení.

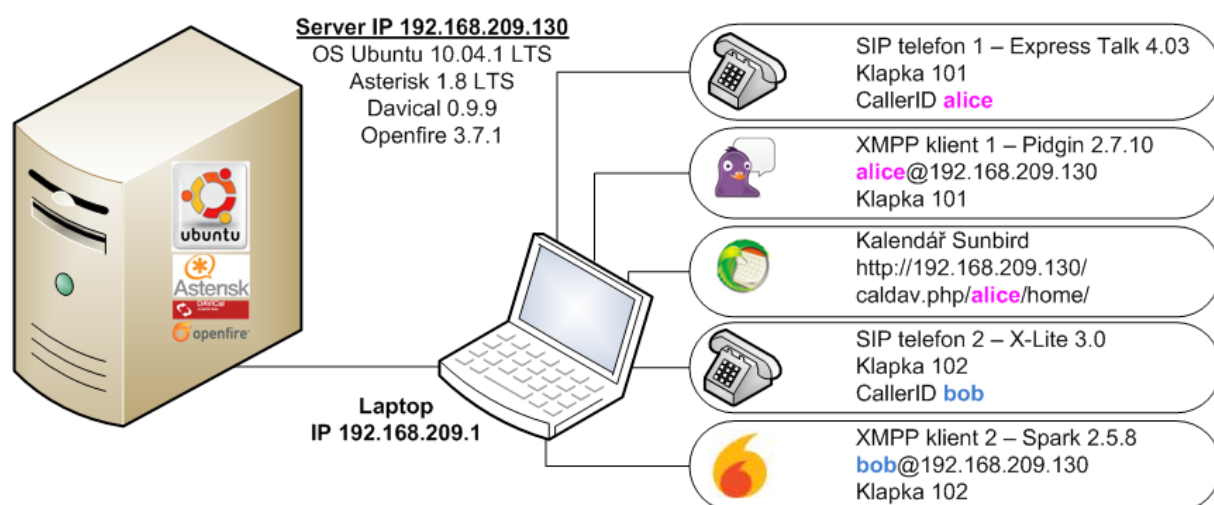
## 3 Praktická část diplomové práce

Tato kapitola obsahuje praktické zapojení, konfiguraci a otestování funkcí, které Asterisk ve spojení s ostatními servery v kontextu prezence nabízí. Začneme instalací operačního systému Ubuntu 10.04.1 na virtuálním stroji, poté provedeme instalaci telefonní ústředny Asterisk, kalendářového serveru Davical, databázového serveru PostgreSQL, XMPP serveru Openfire a klientů využívajících služeb těchto serverů. Po ověření správného zapojení jsou vysvětleny a ověřeny funkce číslovacího plánu.

### 3.1 Schéma zapojení a použité aplikace

#### 3.1.1 Schéma zapojení

Celá praktická část diplomové práce byla realizována na osobním laptopu. K realizaci serveru byl použit virtualizační nástroj VMware Workstation verze 6.5.3. Díky tomuto řešení je instalace serveru nezávislá na hardwaru a jednotná pro kohokoliv, kdo by projekt rekonstruoval. Na server byl zvolen operační systém Ubuntu 10.04.1 Server Edition a to z důvodu předchozích zkušeností s tímto systémem a kvalitní podpoře komunity. Dále je zde nainstalován nejnovější Asterisk 1.8 LTS, který byl vydán 26.října 2010, opensource kalendářový systém Davical, ten ke svému běhu vyžaduje PostgreSQL databázový server. Jako XMPP server jsem vybral projekt Openfire, který je velice uživatelsky přívětivý a obsahuje celou řadu doplňků včetně doplňku k propojení Openfire s Asteriskem. V průběhu konfigurace bylo nutné ještě doinstalovat prostředí pro běh Javy (JRE – Java Runtime Environment) a aplikaci pro převod textu na hlas Festival.



Obrázek 5: Schéma zapojení

### 3.1.2 VMware Workstation

Aplikace VMware workstation [14] se řadí mezi elitu na poli virtualizačních nástrojů. Díky jejímu stabilnímu prostředí a možnosti široké konfigurace nastavení virtuálního stroje je optimálním prostředníkem nejen pro testovací účely této práce, ale i pro využití v produkčním prostředí. VMware nabízí možnost přesně definovat, jaké systémové prostředky chceme virtuálnímu serveru přiřadit a umožňuje i tyto prostředky modifikovat. Každému serveru můžeme určit následující parametry:

- Množství rezervované operační paměti RAM
- Velikost diskového prostoru výhradně pro server
- Počet procesorů nebo jader procesoru, které může využít
- Virtuální síťové rozhraní, které bude využívat – je možnost nechat virtuálnímu stroji přiřadit IP adresu z DHCP serveru, vytvořit pouze připojení mezi virtuálním hostujícím strojem, na kterém VMware běží, nebo virtuální stroj schovat za NAT. Tímto dostane privátní IP adresu a do internetu bude vystupovat pod IP adresou hostovaného PC
- Periferní zařízení, které bude smět virtuální stroj používat (CD/DVD mechanika, disketová mechanika, USB řadič, zvuková karta, atd.) v případě USB řadiče dojde k odpojení jednotky z hostujícího stroje a připojení pouze ke stroji virtuálnímu, není tedy možné sdílet například USB Flash Disk mezi hostem a virtuálem.

### 3.1.3 Ubuntu 10.04.1 Server

Operační systém Ubuntu je postavený na kvalitních základech Debianu, který je znám pro svoje robustní serverové instalace - má silnou tradici spolehlivého výkonu a předvídatelného vývoje. Je dostupný ke stažení zdarma a použitá verze 10.04.1 je zároveň LTS verzí (LTS – Long Term Support) s plnou podporou výrobce až do roku 2015. K této distribuci existuje obrovské množství balíčků, které není třeba kompilovat a je možné je jednoduše nainstalovat přes balíčkovací systém. Mezi silné nástroje, které komunita do této distribuce vkládá je propracovaná virtualizace pomocí platformy KVM. Právě virtualizace je v dnešní době velmi uplatňovanou službou v produkčním prostředí hostingových a jiných firem zabývajících se správou serverů. Stejně tak je možné systém umístit do cloudu se všemi výhodami, které cloud computing nabízí, jako je obrovská škálovatelnost a elasticita – umožní uživatelům rychle měnit výpočetní zdroje dle potřeby.

### 3.1.4 Asterisk 1.8

Asterisk je opensource pobočková ústředna napsaná v programovacím jazyku C a běžící na platformách Linux, Windows, Mac OS X, Open BSD nebo Free BSD. Díky otevřenosti zdrojového kódu může na projektu Asterisku pracovat obrovské množství vývojářů. Asterisk je modulární systém, takže je velmi dobře rozšiřitelný o nové funkce, které telekomunikační trh vyžaduje. Navíc si kdokoli může zdrojové kódy upravovat a přizpůsobit tak běh Asterisku svým požadavkům.

Asterisk je privátní pobočková ústředna (PBX – private branch exchange), která mezi sebou propojuje koncová zařízení s jedním nebo několika poskytovateli hlasových služeb. Základní výhodou tedy je, že není třeba pro každé koncové zařízení pronajímat samostatnou linku a koncová zařízení mezi sebou mohou komunikovat zdarma nezávisle na poskytovateli hlasových služeb. S poskytovatelem může být Asterisk propojen analogovou, isdn linkou nebo širokopásmovým připojením, které umožňuje přenášet velké množství hovorů najednou. Stejně jako telekomunikační operátoři je schopen posílat a přijímat telefonní číslo účastníka a na jeho základě směřovat provoz na konkrétní koncové zařízení, které je k dané klapce přihláшено. Stejně tak podporuje běžně dostupné funkce jako čekání na lince, více koncových zařízení pod jedním číslem, přesměrování hovoru a další funkce, které jsou popsány v teoretické části práce.

Většina společností má dvě samostatné sítě, jednu telefonní a druhou počítačovou. VoIP technologie tento rozdíl maže a přenáší hlasový provoz přes IP síť. Největší výhodou je redukce administrativní práce zaměstnanců, kteří se o síť starají. Pokud telefonní síť provozujeme na IP protokolu, ke správě je třeba znát podobné informace jako pro správu počítačové sítě, není třeba mít speciálně školené zaměstnance na telefonní síť.

K Asterisku je možné připojit jak analogové telefony přes analogovou dvojlinku (přes FXS bránu) tak i hardwarové a softwarové telefony využívající některý z následujících protokolů.

Prvním představitelem je H.323 protokol. Je schopen zajistit konferenční hovory obsahující hlasový, obrazový a datový přenos. Asterisk v základní konfiguraci podporu H.323 neobsahuje, je třeba doinstalovat modul asterisk-oh323. Nástupcem protokolu H.323 a v dnešní době nejrozšířenějším protokolem je SIP (Session Initiation Protocol).

SIP je signalizační protokol pro VoIP hovory. Je standardní součástí instalace Asterisku. Svou jednoduchostí syntaxe je podobný protokolům HTTP nebo SMTP. SIP byl původně odeslán do IETF (Internet Engineering Task Force) v únoru roku 1996. V té době byl velmi jednoduchý, a tak trvalo ještě 3 roky než byl po 11 revizích SIP RFC 2543 uveden. Zprvu byl SIP ignorován, trhu vládl protokol H.323, avšak časem, hlavně díky otevřenosti kódu, došlo k výraznému nárůstu popularity



a postupnému ovládnutí trhu. SIP je protokol aplikační vrstvy používající ke komunikaci port 5060. Může být přenášen jak přes TCP tak UDP protokol transportní vrstvy. Je používán k vybudování, modifikaci a ukončení multimediálních relací jako například telefonní hovory přes internet. SIP nepřenáší média mezi koncovými body, k tomu využívá RTP (Real Time Protocol), který používá vysoká čísla portů k přenosu médií (10000 – 20000).

SIP nebyl první a není ani jediný dnes používaný VoIP protokol, ale v současné době má velkou sílu a podporu prodejců hardwaru. Jeho výhodou je nejen široká podpora a flexibilita, ale především jednoduchost. Všichni noví uživatelé a firmy očekávají podporu SIP.

Asi největší technickou překážkou SIPu je vypořádání se s výzvou problému přenosu komunikace skrz NAT (Network address translation). Jelikož SIP zabaluje adresní informace do svých datových rámců a NAT pracuje nejnižší na síťové (network) vrstvě, adresní informace je automaticky změněna a tedy média streamy nemají správnou adresní informaci potřebnou ke správnému připojení skrz NAT. K tomu navíc firewally běžně pracující s NATem nezvažují příchozí média streamy jako část SIP vyjednávání (signalizace) a blokují toto připojení. Přestože nové firewally a Session Border Controllers jsou si SIPu vědomi a umí s ním relativně dobře pracovat, je tohle považováno za velkou nedokonalost SIPu a způsobuje to nekončící problémy síťovým specialistům, kteří potřebují připojit SIP koncové body do existující síťové infrastruktury. Pro komunikaci mezi ústřednami Asterisk je možno použít protokol IAX. Více viz literatura [2] .

IAX je otevřený protokol, který však dodnes není standardem IETF, což se však očekává od novější verze IAX2 v dohledu několika let. IAX je transportní protokol, který využívá jeden UDP port 4569 jak pro signalizaci tak pro přenos. Jeho unikátní vlastností je seskupování několika relací do jednoho datového proudu, což může být obrovská výhoda ve využití šířky pásma pokud posíláme mnoho simultánních kanálů. IAX2 protokol byl vědomě navržen k práci za zařízeními používající NAT. Použitím jednoho UDP portu jak pro signalizaci tak streamy snižuje počet děr ve firewallu na minimum. Díky tomu je považován za jednoho z nejsnadnějších protokolů k implementaci v zabezpečených sítích. Více viz literatura [2].

The Media Gateway Control protokol také pochází z IETF (Internet Engineering Task Force). Ačkoliv jeho nasezení je hojnější, než by kdo očekával, rychle ztrácí půdu pod nohama kvůli protokolům jako SIP nebo IAX. Je definován v RFC 3435. Jeho cílem bylo, aby koncová zařízení byla co nejjednodušší a aby provoz procházel přes hovorové brány a agenty. Na rozdíl od SIP, MGCP používá centralizovaný model. MGCP telefony nemůžou přímo volat jiným MGCP telefonům. Více viz literatura [2].

The Skinny Call Control Protocol je proprietární Cisco protokol pro Cisco VoIP zařízení. Je standardním protokolem pro koncová zařízení v Cisco Manager PBX. Více viz literatura [2].

Proprietární protokol firmy Nortel má pojmenování UNISTIM. Více viz literatura [2].

Chod ústředny se řídí číslovacím plánem (dialplan). Ten je programován v souboru `/etc/asterisk/extensions.conf` a je tvořen 4 základními prvky. Kontext (Context), Pobočka (Extension), Priorita (Priorities) a Aplikace (Application).

Kontext je pojmenovaná skupina poboček, které slouží určitému účelu. Zaručuje oddělení jednotlivých částí číslovacího plánu od sebe, zároveň je však možno tuto interakci naprogramovat. Jednotlivé kontexty jsou vloženy do hranatých závorek a pojmenování je možné pouze pomocí kombinace písmen, číslic, pomlčky a podtržítka.

Pobočka (Extension) je ve světě telekomunikací známa pod představou číselného identifikátoru sloužícího k dosažení určité telefonní linky. Avšak Asterisk vnímá pod tímto pojmem sérii kroků (aplikací), které se mají vykonat po zavolání dané pobočky, která může (ale nemusí) končit voláním určité linky. Syntaxe je `exten => číslo pobočky,priorita,aplikace(parametry)`, příklad: `exten => 123,1,Answer()`.

Priority určují pořadí kroků uvnitř pobočky. Kroky můžeme buďto číslovat po sobě, anebo použít paramter `n`, za který se automaticky dosadí priorita dalšího kroku. Příklad:

- `exten => 123,1,Answer()` ; vyzvednout hovor
- `exten => 123,n,něco dálej`
- `exten => 123,n,Hangup()` ; zavěsit hovor

Aplikace (Applications) určují akci, která se bude v daný krok vykonávat. Existují aplikace, které v kulatých závorkách nevyžadují parametry jako `Answer()` nebo `Hangup()`, většina z aplikací však parametry vyžaduje, např. `Dial(Sip/123)`. Více viz literatura [2].

Stejně jako měly v minulosti opensource projekty pozitivní vliv na rozvoj informačních technologií, tak i Asterisk a ostatní opensource aplikace zásadním způsobem ovlivňují telekomunikační průmysl. Služby, které byly v minulosti zásadou velkých společností a byly draze zaplacený, si dnes může každý zdarma implementovat a přizpůsobit svým vlastním potřebám. Samozřejmě tato svoboda je vykoupena časem a znalostmi, které jsou na uživatele kladeny. Asterisk není jednoduchý systém na konfiguraci, protože existuje tisíce cest, jak dosáhnout výsledku.

Čas ukáže, jestli se Asterisk zařadí po boku jiných úspěšných opensource projektů, které mohou plně konkurovat komerčním řešením.

### 3.1.5 Openfire 3.7.1

Openfire je opensource XMPP server poskytující služby v oblasti online komunikace (IM – instant messaging communication). Více viz literatura [6]. Vlastní XMPP server v síti přináší výhody rychlé a bezpečné komunikace uvnitř sítě, umožňuje kontrolovat přítomnost zaměstnanců a také umožňuje sledovat a směřovat IM provoz svých zaměstnanců. Samozřejmě zde vyvstává otázka ochrany soukromí jednotlivých uživatelů. Tato otázka však není cílena na projekt Openfire, ale spíše na firemní kulturu a přesahuje rámec této práce. Správa těchto uživatelů může být sjednocena s interním LDAP serverem. Důležité je, že IM komunikace přináší rapidní zrychlení výměny zpráv oproti e-mailu nebo volání. Umožňuje také automaticky reagovat na přihlášení uživatele nebo jiné stavy (automatická odpověď při nepřítomnosti).

Openfire je vyvíjen v programovacím jazyku Java, takže je možné jej provozovat na všech možných platformách. Oproti komerčním řešením typu IBM Lotus Sametime nebo Microsoft Office Live Communication Server má otevřený kód a pracuje s otevřeným XMPP komunikačním protokolem formálně nazývaným Jabber. Tento protokol je od roku 2004 standardem pro prezenci a online komunikaci podle IETF. Je podporován předními světovými společnostmi a miliony uživatelů po celém světě. Implementace a správa Openfire aplikace je poměrně jednoduchá a uživatelsky příjemná. Komunikace mezi jednotlivými klienty je zabezpečena na transportní vrstvě přes TLS. Openfire je vyvíjen společností Ignite Realtime, která poskytuje mimo volnou licenci i komerční řešení pro rozsáhlé nasazení a s tímto související podporu. Zde jsou uvedeny výhody a služby, které Openfire server poskytuje.

- používá otevřené standardy (Java, XMPP, SIP)
- je snadný na instalaci a údržbu, kvalitní administrační rozhraní
- umožňuje propojení s externími databázemi (MySQL, PostgreSQL, Oracle, MS SQL Server, IBM DB2)
- umožňuje integraci s OpenLDAP a Active Directory pro centrální správu uživatelů
- nabízí jednoduchou správu uživatelů a skupin využívajících server
- má kvalitní kontrolu provozu a vytížení serveru s možností exportu do PDF souborů
- podporuje tvorbu chatovacích místností
- slouží jako brána do ostatních veřejných sítí (ICQ, a další)

- umožňuje řídit komunikaci, určuje kdo s kým je schopen komunikovat a kdo není
- umí jednoduše přenášet soubory a umožňuje administrátorovi posílat hromadné zprávy
- obsahuje celou řadu doplňků, příkladem může být zobrazení přítomnosti uživatelů na webu, import a export dat, komunikace s pobočkovou ústřednou Asterisk, apod
- může kooperovat s ostatními Openfire servery v cloudu a vyvažovat tak aktuální zátěž na serveru.
- Spojení Openfire a Asterisku umožňuje kontrolovat, který uživatel právě hovoří a není tak plně dostupný k online komunikaci. Zároveň je možné uživateli Openfire serveru přiřadit pobočku z Asterisku a tímto uživatele přímo volat bez znalosti čísla přiřazené pobočky (s podporou XMPP klienta). Pokud se Asterisk přihlásí k serveru jako uživatel, je schopen číst stavy spřátelených účtů, psát jim zprávy a přijímat jejich odpovědi. Na základě těchto funkcí může přizpůsobovat Asterisk své chování.

### 3.1.6 Davical 0.9.9

Davical je server pro sdílení kalendářů. Jedná se o implementaci Caldav protokolu, který byl navržen pro ukládání zdrojových souborů kalendářů ve formátu iCalendar na vzdáleném sdíleném serveru. K těmto kalendářům uživatelé přistupují prostřednictvím kalendářových klientů. V současné době je protokol Caldav standardem dle RFC 4791 a z toho důvodu nalezneme podporu u většiny dnešních kalendářových klientů jako je Mozilla Calendar (Sunbird/Lightning), Evolution, Mulberry, Chandler nebo iCal a iPhone od Apple.

Aplikace je opensource, čili zdarma ke stažení, používá SQL databázi pro ukládání záznamů, obsahuje přívětivé uživatelské rozhraní, kde můžeme definovat práva čtení a psaní do kalendářů pro jednotlivé uživatele systému.

### 3.1.7 Express Talk 4.03 a X-Lite 3.0

Express Talk a X-Lite jsou komerční softwarové telefony podporující SIP protokol. K testovacím účelům postačuje demo licence, která umožňuje základní funkce volání, přidržení hovoru nebo funkci nerušit.

### 3.1.8 Pidgin 2.7.10

Pidgin je opensource klient pro online komunikaci. Jedná se o multiplatformní aplikaci dostupnou pro Windows, Linux i Mac OS X. Zároveň podporuje velké množství komunikačních protokolů včetně protokolu XMPP používaného pro připojení k XMPP serveru Openfire.

### 3.1.9 Spark 2.5.8

Opensource XMPP klient Spark je vyvíjen stejnou společností jako je XMPP server Openfire. Je navržen k plnému využití bezpečnostních funkcí Openfire serveru. Zároveň obsahuje také integraci Asterisku přímo do klienta v podobě možnosti uskutečnit hovor prostřednictvím tlačítka u přidáných uživatelů, kteří mají na Openfire serveru přiřazené číslo pobočky z Asterisku.

### 3.1.10 Sunbird 1.0

Opensource kalendářový program Sunbird pochází z dílny společnosti Mozilla, která má na svědomí například také programy jako Firefox nebo Thunderbird. Nalezneme v něm podporu správy jak lokálních kalendářů tak i síťových kalendářů komunikujících přes protokoly iCalendar, Caldav a Sun Java systém Calendar Server. Více viz literatura [15].

## 3.2 Instalace operačního systému Ubuntu 10.04.1

K provedení instalace budeme potřebovat následující software:

- VMWare Workstation – aktuální používaná verze 6.5.3  
ke stažení zde <http://www.vmware.com/products/workstation/>
- Obraz instalačního cd s Ubuntu 10.04.1 server edition  
ke stažení zde <http://releases.ubuntu.cz/lucid/ubuntu-10.04.1-server-i386.iso>
- Hostující PC, na kterém VMWare bude spuštěn
- SSH klient putty pro snadnější ovládání serveru  
ke stažení zde <http://www.putty.org/>

Postup instalace:

1. V nainstalovaném programu VMware workstation zvolíme: File » New » Virtual Machine

2. Zvolíme typical » vybereme instalaci z disc image, který máme stažený a uložený v PC. VMWare nám automaticky po načtení detekuje Ubuntu Server 10.04.1 operační systém a přizpůsobí tomu i budoucí instalaci.

3. Zadáme Osobní informace, které budou serveru přiřazeny

- i. Celý název počítače
- ii. Uživatel PC
- iii. Heslo s potvrzením

4. Zadáme název Virtuálního stroje, pod kterým bude uložen v PC a cestu, kde bude vytvořen

5. Nastavíme maximum přiděleného diskového prostoru na pevném disku a uložení virtuálního disku v podobě jednoho souboru.

6. Provedeme nastavení síťového připojení virtuálního serveru v programu Virtual Network editor, který je součástí instalace VMware.

Otevřeme si záložku Host Virtual Adapters zvolíme Add a přidáme ze seznamu virtuální adaptér. Potvrdíme a dáme použít. Přepneme do záložky Host virtual subnet mapping a zde u přidávaného adaptéru nastavíme vpravo šipkou

- Subnet 192.168.209.0 – použít
- DHCP Start IP Address 192.168.209.130
- DHCP End IP Address 192.168.209.131
- Nakonec provedeme v záložce NAT restart služby NAT pro daný virtuální adaptér.
- 

7. Pomocí Customize Hardware zkontrolujeme, popř. změníme nastavení virtuálního stroje.

8. V našem případě byly použity následující parametry

- Memory 512 MB
- Hard Disc 8 GB
- CD/DVD – Auto Detect
- Floppy – Auto Detect
- Network Adapter - Custom – Vmnet vytvořený výše
- USB Controller – Present
- Sound Card – Auto Detect (není vyžadována)
- Display - Autodetect
- Processors - 1

9. Zvolíme pokračovat a proběhne nám běžná instalace OS Ubuntu, ve které není třeba nic speciálně nastavovat, vše proběhne automaticky v pořádku. Co se týče vytváření jednotlivých částí disku, tak instalace nabídne nastavení jednoho hlavního oddílu ve formátu ext4 a jednoho swap oddílu, které můžeme zapsat na virtuální disk. (viz příloha 1)

10. Po dokončení instalace a restartu virtuálního serveru provedeme přihlášení do nového systému pomocí přihlašovacího jména a hesla, které jsme zadali na počátku instalace

11. V terminálové konzoli získáme nejprve IP adresu z DHCP serveru a poté zjistíme, jakou IP adresu dostal server přidělenou, zdali je aktivní připojení na internet, nastavíme časovou zónu a systémový čas. V konzoli použijeme příkazy

```
mik560@ubuntu:~$ sudo dhclient
mik560@ubuntu:~$ ifconfig
mik560@ubuntu:~$ ping downloads.asterisk.org
mik560@ubuntu:~$ ln -sf /usr/share/zoneinfo/Europe/Prague
/etc/localtime
mik560@ubuntu:~$ sudo ntpdate tak.cesnet.cz
```

12. Z důvodu používání vzdáleného přístupu k serveru a výhod z toho plynoucích si provedeme instalaci Openssh serveru.

```
mik560@ubuntu:~$ sudo apt-get update
mik560@ubuntu:~$ sudo apt-get install openssh-server
```

13. Nyní se přihlásíme k serveru pomocí programu putty nebo můžeme pokračovat v práci v konzoli. Pro přihlášení k serveru použijeme IP adresu zobrazenou pomocí ifconfig a port 22. Po vzájemné výměně certifikátu se putty přihlásí na server.

### 3.3 Instalace softwarové ústředny Asterisk

14. Zde se přemístíme do složky /usr/src/ stáhneme instalační soubory Asterisku a provedeme kompilaci na virtuálním stroji. Podrobněji viz literatura [7].

```
mik560@ubuntu:~$ cd /usr/src/
mik560@ubuntu:~$ sudo wget
http://downloads.asterisk.org/pub/telephony/asterisk/asterisk-1.8-
current.tar.gz
mik560@ubuntu:/usr/src$ sudo gzip -d asterisk-1.8-current.tar.gz
mik560@ubuntu:/usr/src$ sudo tar -xf asterisk-1.8-current.tar
mik560@ubuntu:/usr/src$ cd asterisk-1.8.1/
mik560@ubuntu:/usr/src/asterisk-1.8.1$ sudo
./contrib/scripts/install_prereq install
```

během instalace zadáme ITU-T kod pro CR +420

```
mik560@ubuntu:/usr/src/asterisk-1.8.1$  
sudo ./contrib/scripts/get_mp3_source.sh  
mik560@ubuntu:/usr/src/asterisk-1.8.1$ sudo ./configure
```

Pokud nám instalace proběhne bez problému, ukáže se nám ve výpisu logo asterisku a pod ním text:

```
configure: Package configured for:  
configure: OS type      : linux-gnu  
configure: Host CPU     : i686  
configure: build-cpu:vendor:os: i686 : pc : linux-gnu :  
configure: host-cpu:vendor:os: i686 : pc : linux-gnu :
```

Následuje instalace doplňků Asterisku:

```
mik560@ubuntu:/usr/src/asterisk-1.8.1$ sudo make menuselect  
zvolíme  
app_mysql  
cdr_mysql  
format_mp3  
extra-sounds-EN-GSM
```

```
mik560@ubuntu:/usr/src/asterisk-1.8.1$ sudo make install  
mik560@ubuntu:/usr/src/asterisk-1.8.1$ sudo make samples  
mik560@ubuntu:/usr/src/asterisk-1.8.1$ sudo make config
```

Zkontrolujeme stav Asterisku po instalaci

```
mik560@ubuntu:/usr/src/asterisk-1.8.1$ sudo /etc/init.d/asterisk  
status  
Usage: /etc/init.d/asterisk {start|stop|restart|reload|force-reload}
```

Asterisk po instalaci používá několik adresářů v linuxovém systému. Konfigurace těchto cest ke složkám probíhá v souboru `/etc/asterisk/asterisk.conf`. Podrobněji viz literatura [3].

- `/etc/asterisk/` obsahuje konfigurační soubory Asterisku.
- `/usr/lib/asterisk/modules/` obsahuje všechny moduly, které jsme schopni do Asterisku přidat. Jedná se o nejrozumnější aplikace, kodeky, formáty souborů a kanály, kterými může Asterisk komunikovat. Asterisk automaticky tyto moduly načítá po startu.
- `/var/lib/asterisk/` obsahuje soubor `astdb`, který je databází Asterisku (obdobou Microsoft Windows Registry) a celou řadu podadresářů jako `agi-bin`, kam umísťujeme vlastní `agi` skripty, `firmware`, `images`, `keys`, kde nalezneme soukromé a veřejné klíče uživatelů, `mohmp3` pro uložení hudby na pozadí (`music on hold`) a `sounds` obsahující veškeré zvukové



stopy, které Asterisk pro komunikaci s volajícím používá v aplikacích jako hlasový průvodce apod.

- `/var/spool/asterisk/` a jeho podsložky slouží k ukládání záznamů z aplikací diktafon, konferenční hovory nebo hlasová schránka. Zajímavou složkou je složka `outgoing`, pokud do ní vložíme správně strukturovaný soubor, uskuteční se volání dle programu, který soubor obsahuje. Prakticky tak můžeme soubory vkládat cronem v určený čas a programovat tak chování ústředny. Ve složce `monitor` nalezneme nahrávky hovorů, pokud máme nahrávání aktivováno. Složky `system` a `tmp` jsou systémové pro ukládání dočasných souborů.
- `/var/run/` obsahuje číslo procesu pro všechny procesy systému, včetně Asterisku.
- `/var/log/asterisk/` obsahuje logy. Typ informací, které se do logu budou zapisovat můžeme nastavit editací souboru `/etc/asterisk/logger.conf`
- `/var/log/asterisk/cdr-csv` obsahuje podrobné informace o hovorech ve formátu CSV.

### 3.3 Instalace kalendářového serveru Davical

Jakmile máme instalaci Asterisku hotovou, přejdeme k instalaci Davical. Více viz literatura [10].

```
mik560@ubuntu:~$ sudo apt-get install davical
```

Tento software ke svému běhu potřebuje následující knihovny a balíčky, které si během instalace stáhne: `apache2-mpm-prefork` `apache2-utils` `apache2.2-bin` `apache2.2-common` `davical` `libapache2-mod-php5` `libaprutil1-dbd-sqlite3` `libaprutil1-ldap` `libawl-php` `libdbd-pg-perl` `libdbi-perl` `libnet-daemon-perl` `libplrpc-perl` `libyaml-perl` `libyaml-syck-perl` `php5` `php5-common` `php5-pgsql` `postgresql-client-8.4` `postgresql-client-common` `ssl-cert`.

Davical využívá k ukládání údajů databázi PostgreSQL. Tento databázový server si stáhneme a nakonfigurujeme k použití. Podrobněji viz literatura [9].

```
mik560@ubuntu:~$ sudo apt-get install postgresql postgresql-contrib  
php5-pgsql libapache2-mod-auth-pgsql
```

Při instalaci nám vznikl v systému nový uživatel s názvem `postgres`. Tento uživatel je nastaven jako defaultní uživatel databáze PostgreSQL, čili první přihlášení provedeme přes něj. Abychom se ale mohli přihlásit, potřebujeme znát heslo k uživateli `postgres`. Toto heslo však není nastaveno, a proto si ho nastavíme:

```
mik560@ubuntu:~$ sudo passwd -d postgres
mik560@ubuntu:~$ sudo passwd postgres
Enter new UNIX password: mik560
Retype new UNIX password: mik560
passwd: password updated successfully
```

```
mik560@ubuntu:~$ sudo /etc/init.d/postgresql-8.4 restart
```

### Konfigurace Davical databáze

```
mik560@ubuntu:~$ su postgres
postgres@ubuntu:/home/mik560$ sudo
nano/etc/postgresql/8.4/main/pg_hba.conf
```

editujeme soubor dle vzoru – výsledkem je, že uživatel databáze davical má přístup z lokálního stroje.

```
# TYPE      DATABASE      USER                CIDR-ADDRESS        METHOD
# "local" is for Unix domain socket connections only
local       davical       davical_app         trust
local       davical       davical_dba         trust
# IPv4 local connections:
host        all           all                 127.0.0.1/32        md5
# IPv6 local connections:
host        all           all                 ::1/128             md5
```

Pomocí CTRX+X a YES uložíme editovaný soubor v nano editoru.

```
postgres@ubuntu:/home/mik560$ sudo /etc/init.d/postgresql-8.4
restart
```

### Nainstalujeme databázi Davical serveru do PostgreSQL databáze

```
postgres@ubuntu:/home/mik560$ /usr/share/davical/dba/create-
database.sh
```

Výsledkem je zpráva o úspěšné instalaci a heslo pro prvotní přihlášení do systému Davical. Dříve než se přihlásíme, musíme ještě aplikaci zpřístupnit online konfigurací virtualhostu v Apache2. Nejprve však odhlásíme uživatele postgres příkladem CTRL+D. Do defaultního virtual hostu Apache přidáme následující záznam.

```
mik560@ubuntu:/home/mik560$ sudo nano /etc/apache2/sites-
enabled/000-default
```

```
# Virtual Host for DAViCal
<VirtualHost 192.168.209.130 >
    DocumentRoot /usr/share/davical/htdocs
    DirectoryIndex index.php index.html
    ServerName 192.168.209.130
    ServerAlias 192.168.209.130
```

```

Alias /images/ /usr/share/davical/htdocs/images/
<Directory /usr/share/davical/htdocs/>
    AllowOverride None
    Order allow,deny
    Allow from all
</Directory>
AcceptPathInfo On
php_value include_path /usr/share/awl/inc
php_value magic_quotes_gpc 0
php_value register_globals 0
php_value error_reporting "E_ALL & ~E_NOTICE"
php_value default_charset "utf-8"
</VirtualHost>

```

```
mik560@ubuntu:/home/mik560$ sudo /etc/init.d/apache2 restart
```

Davical server vyžaduje editaci konfiguračního souboru dle vzoru níže:

```

mik560@ubuntu:/home/mik560$ sudo nano /etc/davical/192.168.209.130-
conf.php
<?php
// $c->domain_name = "calendar.example.net";
// $c->sysabbr      = 'DAViCal';
// $c->admin_email  = 'admin@example.net';
// $c->system_name  = "Example DAViCal Server";
// $c->enable_row_linking = true;
$c->pg_connect[] = 'dbname=davical port=5432 user=davical_app';
?>

```

Nyní je náš kalendářový systém zprovozněn a nalezneme jej na URL adrese serveru <http://192.168.209.130/>, kterou zadáme do prohlížeče. Přihlásíme se pod uživatelem admin heslo XXXXXXXXX, které se nám vygenerovalo během instalace.

Vytvoříme si zde kalendář, který bude pravidelně sledován asteriskem.

<http://192.168.209.130/usr.php?create>

```

User Name:      asterisk
New Password:   asterisk
Confirm:        asterisk
Full Name:      asterisk
Email:          asterisk@localhost
Active:         ano
User Roles:     všechny ano

```

Create

Kalendář nyní bude dostupný pod URL <http://192.168.209.130/caldav.php/asterisk/home/>, login a heslo nastaveno výše. Pokud máte problém s přihlášením, proveďte přenastavení hesla.

Tyto údaje můžete použít v jakémkoliv kalendářovém klientu, který podporuje protokol caldav, příkladem budiž opensource program Sunbird.

## 3.4 Instalace XMPP serveru Openfire

Přistoupíme k instalaci openfire. Instalovat budeme z balíčku, který si stáhneme ze stránek výrobce softwaru.

```
mik560@ubuntu:~$ sudo mkdir /usr/src/openfire
mik560@ubuntu:~$ cd /usr/src/openfire/
mik560@ubuntu:/usr/src/openfire$ sudo wget
http://www.igniterealtime.org/downloadServlet?
filename=openfire/openfire_3.7.0_all.deb
```

Openfire vyžaduje ke svému běhu java6-jre, proto ji musíme do systému doinstalovat. Bohužel sun-java6-jre není součástí běžných balíčků Ubuntu Lucid, je tedy třeba rozšířit repozitáře. Více viz literatura [11].

```
mik560@ubuntu:/usr/src/openfire$ sudo apt-get install python-
software-properties
mik560@ubuntu:/usr/src/openfire$ sudo add-apt-repository "deb
http://archive.canonical.com/ lucid partner"
mik560@ubuntu:/usr/src/openfire$ sudo apt-get update
mik560@ubuntu:/usr/src/openfire$ sudo -i
root@ubuntu:~# apt-get install sun-java6-jre
```

Během instalace je třeba potvrdit licenční podmínky Sun, použijte F12 a yes.

```
root@ubuntu:~# cd /usr/src/openfire
root@ubuntu:/usr/src/openfire$ dpkg -i downloadServlet\?
filename=openfire%2Fopenfire_3.7.0_all.deb
root@ubuntu:~# /etc/init.d/openfire start
root@ubuntu:~# su postgres
postgres@ubuntu:/root$ createuser openfire
Shall the new role be a superuser? (y/n) n
Shall the new role be allowed to create databases? (y/n) y
Shall the new role be allowed to create more new roles? (y/n) y
postgres@ubuntu:/root$ createdb openfire
```

```
postgres@ubuntu:/root$ psql openfire
openfire=# ALTER USER openfire with password 'openfire';
openfire=# ALTER DATABASE openfire OWNER TO openfire;
```

Instalaci dokončíme ve webovém rozhraní na URL adrese

<http://192.168.209.130:9090/setup/index.jsp>

```
Choose Language:          Czech (cs_CZ)
Doména:                   192.168.209.130
Port administrátorské konzole: 9090
Zabezpečený port administrátorské konzole: 9091
Nastavení databáze:       Standardní databázové
                             připojení
Předvolby databázového ovladače: PostgreSQL
URL databáze: jdbc:postgresql://localhost:5432/openfire
Uživatelské jméno:         openfire
Heslo:                     openfire
Nastavení profilu:         Výchozí - Ukládat
                             uživatele a skupiny v
                             databázi Openfire.

Administrátorský účet:     root@localhost
Nové heslo:                mik560
Potvrzení hesla:          mik560
```

Nyní se můžeme přihlásit do administrátorské konzole

<http://192.168.209.130:9090/login.jsp> admin / mik560

zde provedeme seřízení časové zóny

<http://192.168.209.130:9090/server-locale.jsp> Central European Time (1 GMT)

Nyní bude následovat propojení periferních zařízení a softwarových klientů s jednotlivými službami na serveru.

## 3.5 Propojení Asterisk a Davical serveru

Pro komunikaci budu používat program Sunbird instalovaný na PC

Ke stažení je zde <http://www.mozilla.org/projects/calendar/sunbird/>

Po instalaci program spustíme a nastavíme dle návodu:

```
File > New calendar > On the Network
Format: CalDAV
Location: http://192.168.209.130/caldav.php/asterisk/home/
Name: asterisk
```

```
Show Alarms: ano
Finish
```

Zadejte požadované přihlašovací údaje ke kalendáři: asterisk / asterisk, pokud by Sunbird přihlášení nevyžadoval a nechtěl se s kalendářem spojit, použijte URL kalendáře ve tvaru

```
Location:
http://asterisk:asterisk@192.168.209.130/caldav.php/asterisk/home/
```

Nyní provedeme nastavení Asterisku, aby mohl číst a zapisovat do vytvořeného kalendáře asterisk na caldav serveru. Na straně Asterisku si editujete soubor calendar.conf

```
mik560@ubuntu:/usr/src/openfire$ sudo nano
/etc/asterisk/calendar.conf
```

```
[asterisk]
type = caldav
url = http://192.168.209.130/caldav.php/asterisk/home/
user = asterisk
secret = asterisk
refresh = 1
```

uložíme a provedeme restart Asterisku

```
mik560@ubuntu:/usr/src/openfire$ sudo /etc/init.d/asterisk restart
```

Stejným způsobem Asterisk komunikuje s kalendáři ve formátu ical a Microsoft Exchange Serverem 2003 a 2007.

## 3.6 Připojení XMPP klientů

XMPP serveru slouží k Instant Messaging (dále IM) komunikaci mezi IM klienty podporující protokol XMPP. Pro účely testování jsem zvolil XMPP klienty Pidgin, dostupný pro většinu operačních systémů a Spark, který je vyvíjen stejnou společností jako je XMPP server Openfire. Začneme s instalací Pidgin klienta. Software je dostupný ke stažení zde: <http://www.pidgin.im/download/>, po instalaci zvolte účty > spravovat > přidat a nastavte (obrázek viz příloha 2)

- Protokol: XMPP
- Jméno: alice
- Doména: 192.168.209.130
- Heslo: alice
- Vytvořit tento nový účet na serveru: ANO

Po provedení příkazu přidat se nám objeví autorizace ze strany serveru, doplníme údaje: (viz příloha 3).

- Username: alice
- Full Name: alice
- Email: alice@localhost
- Password: alice

Po vložení požadovaných údajů dostaneme potvrzení o úspěšné registraci. (viz příloha 4)

Zvolíme v pidginu účet aktivovat a server se nás ještě jednou zeptá na heslo k účtu, které dáme pro zjednodušení zapamatovat.

Stejně tak nastavíme i druhého XMPP klienta vytvořeného stejnou společností jako Openfire. Jedná se XMPP klienta Spark. Je k dispozici ke stažení na URL <http://www.igniterealtime.org/projects/spark/index.jsp>

Po instalaci vytvoříme nový účet pomocí Accounts. Vyplníme požadované údaje: (viz příloha 5)

- Username: bob
- Password: bob
- Confirm Password: bob
- Server: 192.168.209.130

Dostaneme potvrzení: New account has been created. Automaticky se nám vyplní přihlašovací údaje a zvolíme Login. Nyní si přidáme Alici jako kamaráda, abychom mohli spolu komunikovat.

```
Contact > Add Contact >  
UserName: Alice
```

Na pidgin nám dorazí zpráva o autorizaci kamaráda, kde zvolíme autorizovat. Odklikneme přidání.

## 3.7 Připojení SIP klientů

Nyní vytvoříme Alici a Bobovi telefonní účty na Asterisku. Provedeme to přes konzoli v Ubuntu. Na konec souboru přidáme konfiguraci dle návodu.

```
mik560@ubuntu:~$ sudo nano /etc/asterisk/sip.conf  
[101]  
type=friend  
regexten=101 ; pobočka pod kterou bude uživatel  
dostupný  
callerid="alice" <101> ; jméno, které se zobrazí na displeji
```

```

host=dynamic                ; host se může přihlásit z jakékoliv
IP adresy
nat=yes
directmedia=no
disallow=all
allow=ulaw
allow=alaw
mailbox=6001@default        ; hlasova posta
registertrying=yes

```

```

[102]
type=friend
regexten=102                ; pobočka pod kterou bude uživatel
dostupný
callerid="bob" <102>        ; jméno, které se zobrazí na displeji
host=dynamic                ; host se může přihlásit z jakékoliv
IP adresy
nat=yes
directmedia=no
disallow=all
allow=ulaw
allow=alaw
mailbox=102@default
registertrying=yes

```

Heslo k účtu nebudeme nyní pro zjednodušení nastavovat, jinak se nastavuje zde řádkem `secret=heslo`.

Uložíme a nastavíme číslovací plán, opět přidáme text na konec souboru

```

mik560@ubuntu:~$ sudo nano /etc/asterisk/extensions.conf
[custom]
exten=>101,1,Dial(SIP/101,20)
exten=>102,1,Dial(SIP/102,20)
a na řádek 666 připojíme nově vytvořený kontext do defaultního kontextu.

```

```

[default]
include => custom

```

Po uložení úprav restartujeme Asterisk

```

mik560@ubuntu:~$ sudo /etc/init.d/asterisk restart

```

Pro komunikaci budu používat softwarový telefon X-lite pro Alici a pro Boba Express talk. Ke stažení je software dostupný v příloze na CD nebo zde:

<http://www.nch.com.au/components/talksetup.exe>

[http://download.chip.eu/cz/X-Lite-3.0\\_117015.html](http://download.chip.eu/cz/X-Lite-3.0_117015.html)



Po instalaci X-lite zvolíme pravé tlačítko a Sip Account Settings > Properties a nastavíme pobočku Alice (viz příloha 6).

- Display Name: alice
- User name: 101
- Password: alice
- Authorization user name: 101
- Domain: 192.168.209.130

Po registraci k ústředně bude telefon ve stavu ready s číslem přihlášené pobočky. (viz příloha 7)

Podobné nastavení bude u Express Talk, po instalaci zvolíme Options > Lines. Obrázek tohoto nastavení naleznete v příloze č.8.

- Full 'Friendly' Display Name: Bob
- Server (SIP Proxy or Virtual PBX): 192.168.209.130
- SIP Number (or User Name): 102
- Password: bob

Po přihlášení uvidíme dole v logu telefonu potvrzovací zprávu

Restarting so setting take efect

Attempting to register sip:102@192.168.209.130

Registered as: sip:102@192.168.209.130

## 3.8 Propojení Asterisku a Openfire serveru

V Asterisku povolíme přihlášení přes AMI (asterisk manager interface)

```
mik560@ubuntu:~$ sudo nano /etc/asterisk/manager.conf
[general]
```

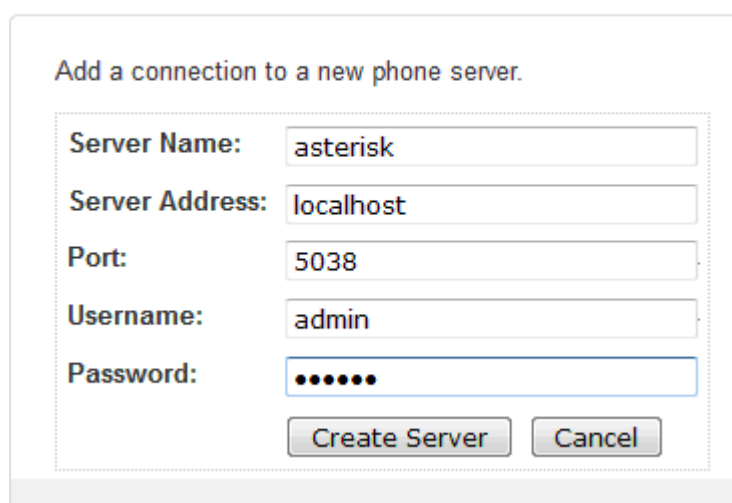
```
enabled = yes
port = 5038
bindaddr = 0.0.0.0
[admin]
secret = mik560
all
read =
system,call,log,verbose,agent,user,config,dtmf,reporting,cdr,dialplan
write = system,call,agent,user,config,command,reporting,originate
```

V openfire přidáme plugin pro komunikaci s Asterisk serverem.

<http://192.168.209.130:9090/login.jsp>

Doplňky > Dostupné doplňky > Asterisk-IM Openfire Plugin > Instalovat. Zvolíme General Settings a aktivujeme tento plugin pomocí volby Enabled. Volbu uložíme tlačítkem save, což nám aktivuje volbu Add Server. Zde doplníme nastavení tak, jak jsme jej nastavili v asterisku v manager.conf. Heslo je mik560.

## Create Phone Server



Add a connection to a new phone server.

Server Name: asterisk

Server Address: localhost

Port: 5038

Username: admin

Password: •••••

Create Server Cancel

Obrázek 6: Propojení Asterisk a Openfire serverů

Úspěšné připojení nám deklaruje zelená ikona

Name	Address	Port	Username
● asterisk	localhost	5038	admin

Obrázek 7: Kontrola funkčnosti propojení Asterisk a Openfire serverů

Nyní přiřadíme XMPP účtům telefonní pobočky. Zvolte Phone Mapping a nastavte dle obrázku.

Username	Device	Extension
bob	SIP/102	102
alice	SIP/101	101

Obrázek 8: Přiřazení klapky z Asterisku XMPP uživatelům

Dále provedeme přihlášení XMPP klienta Asterisk k Openfire serveru. Asterisk po přihlášení bude vyžadovat potvrzení přátelství ze strany účtů `alice@192.168.209.130` a `bob@192.168.209.130`. Po dokončení autorizace bude schopen uživatel Asterisk číst stav těchto dvou uživatelů, psát jim zprávy a číst jejich odpovědi a na základě těchto informací přizpůsobovat chování ústředny.

```
mik560@ubuntu:~$ sudo nano /etc/asterisk/jabber.conf
[asterisk]
type=client                                ; přihlášení jako klient
nebo komponenta
serverhost=192.168.209.130                 ; cesta kXMPP serveru
username=asterisk@192.168.209.130          ; uživatelské jméno XMPP
klienta
secret=asterisk                            ; heslo
port=5222                                  ; Port defaultně 5222
buddy=bob@192.168.209.130                 ; seznam známých uživatelů
buddy=alice@192.168.209.130
distribute_events=no
status=available                           ; Jeden z: chat, available,
away, xaway, or dnd
statusmessage="I am available"             ; Text stavu Asterisku
timeout=5
```

Po uložení a restartování Asterisku se nám XMPP klient Asterisk autorizuje k uživatelům Alice a Bob, autorizaci schválíme.

```
mik560@ubuntu:~$ sudo /etc/init.d/asterisk restart
```

Od této chvíle bude Asterisk informovat Openfire o tom, že daný uživatel má hovor, Openfire tak jeho status změni na „On Phone“, což odpovídá stavu Away (Pryč). Zároveň je možno ze SIP klienta Spark vytočit uživatele, který má pobočku přidělenou. (viz příloha č.9) Po uskutečnění hovoru zavolá Asterisk volajícího a po vyvěšení telefonu jej spojí s volaným. Není tak potřeba pamatovat si jednotlivé čísla účastníků a vést tak samostatný adresář.

## 3.9 Aplikace funkcí Jabberu

### 3.9.1 JabberSend()

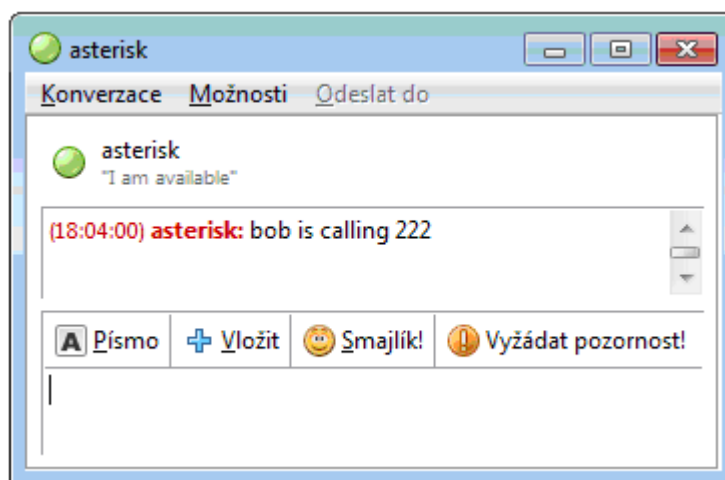
Funkce JabberSend je funkce, která odešle text XMPP uživateli prostřednictvím XMPP uživatele Asterisk přihlášeného k XMPP serveru na localhostu.

Na konec souboru extension.conf přidáme pobočku 222, která nám vypíše do okna kdo a jakou pobočku volá.

```
mik560@ubuntu:~$ sudo nano /etc/asterisk/extensions.conf

exten => 222,1,JabberSend(asterisk,alice@192.168.209.130,${
{CALLERID(name)} is calling ${EXTEN});
exten => 222,n,Dial(SIP/101, 30);
exten => 222,n, Hangup();
```

Po uložení a reloadu Asterisku otestujeme hovor z klienta 102 (Bob – Express Talk) na číslo 222. V okně Pidginu, který používá Alice uvidíme stav na obrázku.



Obrázek 9: Zaslání zprávy z Asterisků do okna XMPP klienta

### 3.9.2 JabberReceive()

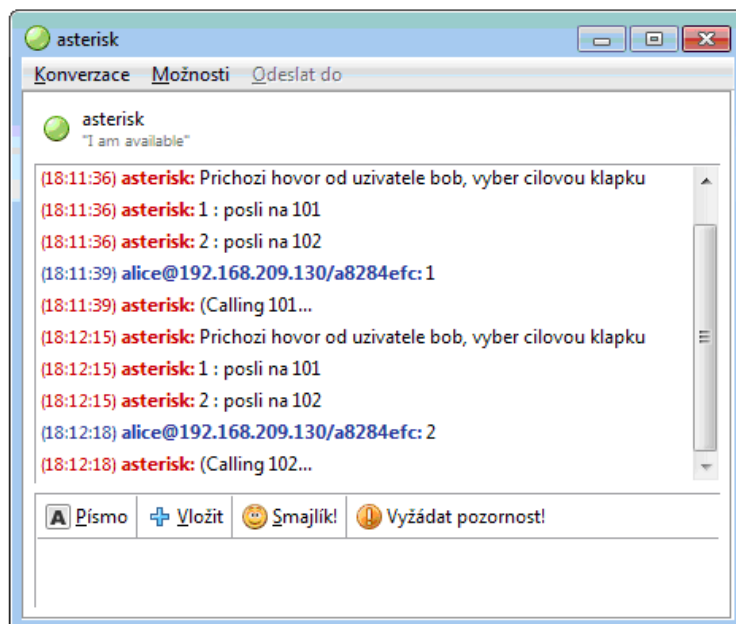
Funkci JabberSend rozšíříme o funkci Jabber\_Receive a umožníme uživateli vybrat, kam má přichodzí hovor nasměrovat. Na konec souboru přidáme pobočku 333, která oznámí volanému, že mu

volá určitá osoba a nabídne mu možnost směřovat telefon buďto na pobočku 101 nebo na pobočku 102.

```
mik560@ubuntu:~$ sudo nano /etc/asterisk/extensions.conf
```

```
exten => 333,1,Answer();
exten => 333,n,
JabberSend(asterisk,alice@192.168.209.130,Prichodzi hovor od
uzivatele ${CALLERID(name)}, vyber cilovou pobočku);
exten => 333,n,          JabberSend(asterisk,alice@192.168.209.130,1 :
posli na 101);
exten => 333,n,          JabberSend(asterisk,alice@192.168.209.130,2 :
posli na 102);
exten => 333,n,          Set(OPTION=$
{JABBER_RECEIVE(asterisk,alice@192.168.209.130)});
exten => 333,n,          GotoIf($[${OPTION} = 1]?10:20)
exten => 333,10,          JabberSend(asterisk,alice@192.168.209.130,
(Calling 101...);
exten => 333,n,          Dial(SIP/101);
exten => 333,20,          JabberSend(asterisk,alice@192.168.209.130,
(Calling 102...);
exten => 333,n,          Dial(SIP/102);
```

Po uložení a reloadu asterisků otestujeme hovor z uživatele 102 (Bob – Express Talk) na číslo 333. V okně Pidginu, který používá Alice uvidíme stav na obrázku. Nejdříve volím 1 a hovor je směřován na 101. Poté testuji znovu a volbou 2 je hovor směřován na linku 102.



Obrázek 10: Komunikace Asterisků s XMPP uživatelem

### 3.9.3 JabberStatus()

Jabber\_Status funkce zjistuje stav určeného XMPP uživatele a na základě tohoto stavu provádí nejrůznější úkony.

Funkce vrací následující stavy:

- 1 Přítomen
- 2 Volný k hovoru ( Free to chat)
- 3 Pryč
- 4 Pryč na dlouho
- 5 Nerušit
- 6 Offline (nebylo navozeno, vracel vždy 7)
- 7 cokoliv jiného (odhlášen)

V tomto skriptu jsou hovory podle stavu směrovány na různé linky. Na konec souboru přidáme pobočku 444.

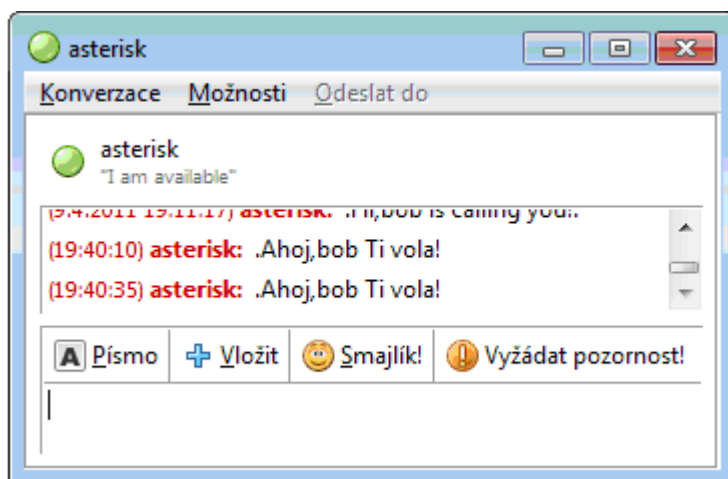
```
mik560@ubuntu:~$ sudo nano /etc/asterisk/extensions.conf
```

```
exten => 444,1,Set (STATUS=${
{JABBER_STATUS(asterisk,bob@192.168.209.130)});
exten => 444,n,GotoIf ($[${STATUS} = 1]?10:20)
exten => 444,10,Dial (SIP/101)
exten => 444,n,Hangup()
exten => 444,20,GotoIf ($[${STATUS} = 2]?21:30)
exten => 444,21,Dial (SIP/102)
exten => 444,n,Hangup()
exten => 444,30,GotoIf ($[${STATUS} = 3]?31:40)
exten => 444,31,Dial (SIP/103)
exten => 444,n,Hangup()
exten => 444,40,GotoIf ($[${STATUS} = 4]?41:50)
exten => 444,41,Dial (SIP/104)
exten => 444,n,Hangup()
exten => 444,50,GotoIf ($[${STATUS} = 5]?51:60)
exten => 444,51,Dial (SIP/105)
exten => 444,n,Hangup()
exten => 444,60,GotoIf ($[${STATUS} = 6]?61:70)
exten => 444,61,Dial (SIP/106)
exten => 444,n,Hangup()
exten => 444,70,GotoIf ($[${STATUS} = 7]?71:80)
exten => 444,71,Dial (SIP/107)
exten => 444,n,Hangup()
exten => 444,80,Busy()
exten => 444,n,Hangup()
```

A další skript nám zjistí, zdali je volaný uživatel dostupný nebo ne a podle toho směřuje hovor a informuje volajícího i volaného.

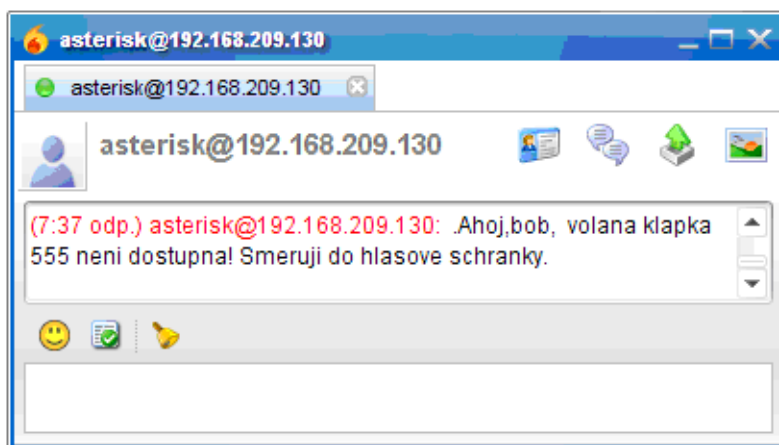
```
exten => 555,1,Set(STATUS=${
{JABBER_STATUS(asterisk,alice@192.168.209.130)}});
exten => 555,2,gotoif($[${STATUS}] < 7)?3:10)
exten => 555,3(available),jabbersend(asterisk,alice@192.168.209.130,
.Ahoj,${CALLERID(name)} Ti vola!)
exten => 555,4,Dial(SIP/101)
exten => 555,10,jabbersend(asterisk,$
{CALLERID(name)}@192.168.209.130, .Ahoj,${CALLERID(name)}, volana
pobočka ${EXTEN} není dostupna! Smeruji do hlasove schranky.)
exten => 555,11(unavailable),Dial(SIP/102)
```

V případě, že je Alice dostupná, zobrazí se v jejím okně zpráva o volání a zvoní jí telefon.



Obrázek 11: Oznámení o příchozím volání

V případě nepřítomnosti dostane zprávu volající Bob



Obrázek 12: Oznámení o nedostupnosti volaného

## 3.10 Aplikace kalendářových funkcí

### 3.10.1 CALENDAR\_BUSY()

CALENDAR\_BUSY() funkce nám vrátí aktuálně nastavenou zaneprázdněnost, kterou má uživatel nastavenou v kalendáři. Tento stav kontroluje dle nastaveného limitu v /etc/asterisk/calendar.conf. Na konec souboru extension.conf přidáme pobočku 666.

```
mik560@ubuntu:~$ sudo nano /etc/asterisk/extensions.conf

exten => 666,1,GotoIf(${CALENDAR_BUSY(asterisk)}?busy:free)
exten => 666,n(busy),Dial(SIP/102)
exten => 666,n,Hangup()
exten => 666,n(free),Dial(SIP/101)
exten => 666,n,Hangup()
```

Pokud tedy zavoláme na číslo 666 a účastník bude dle kalendáře dostupný, půjde hovor na pobočku 101, pokud dostupný nebude, hovor půjde na pobočku 102.

### 3.10.2 CALENDAR\_WRITE()

Asterisk umí také do kalendáře zapisovat, jako příklad jsem vytvořil jednoduché logování hovorů. Pokud dojde k vytočení pobočky 777, volá se pobočka 101 a po skončení hovoru se do kalendáře zapíše, kdo a jaký časový interval volal. Tento log si můžeme zkontrolovat ve výpisu událostí na Caldav serveru na níže uvedené URL adrese nebo po reloadu v kalendářovém klientu Sunbird.

[http://192.168.209.130/collection.php?user\\_no=1001&dav\\_name=/asterisk/home/](http://192.168.209.130/collection.php?user_no=1001&dav_name=/asterisk/home/)

```
mik560@ubuntu:~$ sudo nano /etc/asterisk/extensions.conf

exten => 777,1,Set(start=${EPOCH})
exten => 777,n,Dial(SIP/101)
exten => h,1,Set(end=${EPOCH})
exten => h,n,Set(CALENDAR_WRITE(asterisk,summary,start,end)=Volal ${CALLERID(all)},{start},{end})
```



### 3.10.3 CALENDAR\_QUERY()

K dalším funkcím, které Asterisk nabízí je funkce CALENDAR\_QUERY(), který má jako parametry název kalendáře a čas ve formátu unix time. Vrací ID a pole události (event), které použijeme ve funkci CALENDAR\_QUERY\_RESULT. Tato funkce přijímá parametry CALENDAR\_QUERY\_RESULT(id z CALENDAR\_QUERY, požadovaná informace, pořadí události)

Mezi požadované informace patří:

- summary – název události
- description – úplný popis události
- organizer – osoba, která událost organizuje
- location – místo události
- calendar – název kalendáře, ve kterém je událost zaznamenána
- uid – jedinečný identifikátor události
- start – začátek události ve formátu unix time (počet sekund od Unix epoch)
- end – konec události ve formátu unix time (počet sekund od Unix epoch)
- busystate – 0 volný, 1 možná, 2 nepřítomný
- attendees – seznam pozvaných účastníků události oddělených čárkou

Pro přehrání informací je třeba nainstalovat na server Text To Speech aplikaci. Asterisk nativně podporuje aplikaci Festival, ve které najdeme i částečnou českou lokalizaci.

```
mik560@ubuntu:~$ sudo apt-get install festival festival-czech  
festvox-czech-ph
```

na konec souboru zkopírujeme následující konfiguraci [12]

```
mik560@ubuntu:~$ sudo nano /usr/share/festival/festival.scm  
  
;; Enable access to localhost (needed by debian users)  
(set! server_access_list '("localhost\\".localdomain" "localhost"))  
;; (set! voice_default 'voice_pc_diphone)  
(require 'czech)  
(require 'czech-unisyn)  
(set! inhibit-initial-pauses t)  
;; (voice_czech_ph)  
(set! voice_default 'voice_czech_ph)  
;;; Command for Asterisk begin  
(define (tts_textasterisk string mode)  
  (let ((wholeutt (utt.synth (eval (list 'Utterance 'Text string)))))  
    (utt.wave.resample wholeutt 8000)  
    (utt.wave.rescale wholeutt 5)  
    (utt.send.wave.client wholeutt)))  
;;; Command for Asterisk end
```

## Uložíme a propojíme asterisk s aplikací Festival

```
mik560@ubuntu:~$ sudo nano /etc/asterisk/festival.conf
```

```
[general]
host=localhost
port=1314
usecache=yes
cachedir=/var/cache/asterisk/festival/
festivalcommand=(tts_textasterisk "%s" 'file) (quit)\n
```

## Spustíme festival v režimu server

```
mik560@ubuntu:~$ sudo festival --server &
mik560@ubuntu:~$ sudo /etc/init.d/asterisk restart
```

Následující ukázka využití CALENDAR\_QUERY() funkce spojenou s Text To Speech aplikací. Po zavolání na pobočku 888 se dozvíme události, které máme naplánovány v kalendáři na příštích 60 minut.

```
mik560@ubuntu:~$ sudo nano /etc/asterisk/extensions.conf
```

```
exten => 888,1,Answer
exten => 888,n,Set(id=${CALENDAR_QUERY(asterisk,${EPOCH},${[${EPOCH}
+24*60*60]})})
exten => 888,n,Set(num=${CALENDAR_QUERY_RESULT(${id},getnum)})
exten => 888,n,Set(i=1)
exten => 888,n,Festival("Plan na dalsich 60 minut je nasledujici")
exten => 888,n,Festival(Celkovy pocet schuzek je ${num})
exten => 888,n,While(${i} <= ${num})
exten => 888,n,Festival(schuzka cislo ${i} ma zacatek)
exten => 888,n,SayUnixTime(${CALENDAR_QUERY_RESULT(${id},start,${
i})),,Q 'digits/at' Imp)
exten => 888,n,Festival(a konec je v)
exten => 888,n,SayUnixTime(${CALENDAR_QUERY_RESULT(${id},end,${
i})),,Imp)
exten => 888,n,Festival(misto schuzky je: ${CALENDAR_QUERY_RESULT($
{id},location,${i}))})
exten => 888,n,Festival(Schuzka je zapsana v kalendari: $
{CALENDAR_QUERY_RESULT(${id},calendar,${i}))})
exten => 888,n,Festival(Obsah schuzky je: ${CALENDAR_QUERY_RESULT($
{id},summary,${i}))})
exten => 888,n,Set(i=${i} + 1)
exten => 888,n,EndWhile
```

### 3.10.4 CALENDAR\_EVENT()

Podobnou funkcí jako CALENDAR\_QUERY je funkce CALENDAR\_EVENT(). Tuto funkci použijeme pro hlasové oznámení blížící se schůzky 10 minut před jejím začátkem. Vytvoříme pro tento účel kontext calendar\_event\_notify a umístíme jej do dialplanu v souboru extension.conf. Na tento kontext se budeme odkazovat ze souboru calendar.conf.

```
mik560@ubuntu:~$ sudo nano /etc/asterisk/extensions.conf
```

```
[calendar_event_notify]
exten => s,1,Answer
exten => s,n,Festival(za 10 minut mate schuzku\, která se kona v $
{CALENDAR_EVENT(location)}\, obsahem schuzky je $
{CALENDAR_EVENT(description)}\, nashledanou)
exten => s,n,Hangup
```

```
mik560@ubuntu:~$ sudo nano /etc/asterisk/calendars.conf
```

```
[asterisk]
type = caldav
url = http://192.168.209.130/caldav.php/asterisk/home/
user = asterisk
secret = asterisk
refresh = 1
timeframe = 60
autorereminder =
channel = SIP/101
context = calendar_event_notify
extension = s
```

## 3.11 Souhrn všech funkcí – prezenční skript

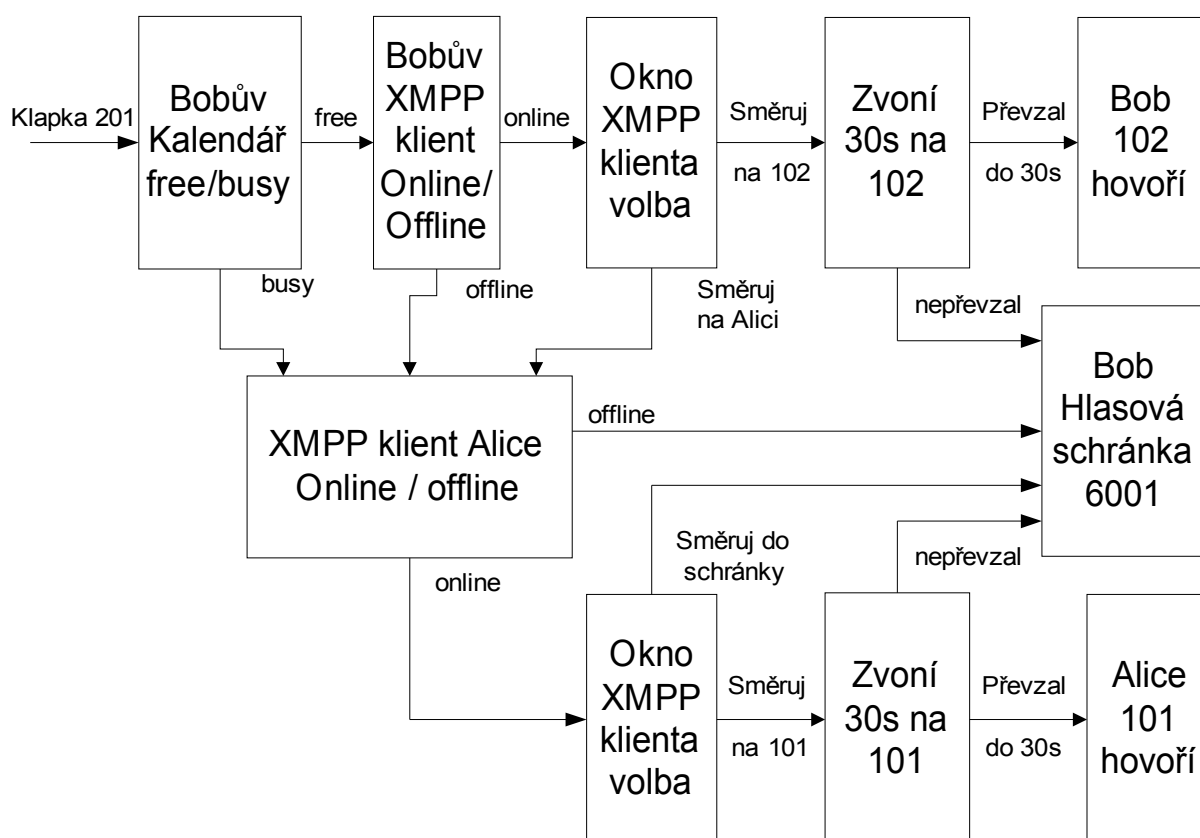
V poslední kapitole praktické části diplomové práce byl vytvořen skript využívající všech funkcí, které byly zmíněny výše. Cílem tohoto skriptu je směřovat hovor na základě stavu v kalendáři a XMPP klientu.

Skript funguje následovně: Uživatel Bob je dostupný na privátní klapce 102. Jeho sekretářka Alice je dostupná na klapce 101. Bob je dostupný na veřejné klapce 201 a má zřízenou voicemailovou schránku pod pobočkovým číslem 6001 pro případ, že by nebral telefon.

Asterisk sleduje Bobův stav v kalendáři a XMPP klientu Spark. Pokud nemá žádnou schůzku a je online, dostane Bob zprávu na XMPP klienta o tom, že mu volá konkrétní uživatel a on má

možnost hovor přijmout (poslat na 102) nebo poslat na Alici. Pokud hovor nevyzvedne, končí hovor v hlasové schránce.

Pokud není Bob dostupný díky kalendáři nebo Sparku, hovor je směrován na Alici na pobočku 101. U Alice se zjišťuje, zdali je přítomna v práci (má aktivního XMPP klienta Pidgin). Pokud ano, zvoní ji telefon a má 30s na vyzvednutí, jinak hovor končí Bobovi v hlasové schránce



Obrázek 13: Schéma chování prezenčního skriptu

```
mik560@ubuntu:~$ sudo nano /etc/asterisk/voicemail.conf
```

```
[mailbox-102]
6001 => 201,bob,bob@localhost
```

```

mik560@ubuntu:~$ sudo nano /etc/asterisk/extensions.conf
;je Bob dostupny v kalendari?
exten => 201,1,GotoIf(${CALENDAR_BUSY(asterisk)}?busy:free)
;ano, Bob je dostupny v kalendari, je dostupny i na chatu? (je u
telefonu?)
exten => 201,n(free),Set(STATUS=$
{JABBER_STATUS(asterisk,bob@192.168.209.130)});
exten => 201,n,gotoif(${[${STATUS}] < 7}?5:busy)
;ano, Bob je dostupny v kalendari i na telefonu
exten => 201,5,
JabberSend(asterisk,bob@192.168.209.130,Prichozi hovor od uzivatele
${CALLERID(name)}, vyber cilovou pobočku);
exten => 201,n,          JabberSend(asterisk,bob@192.168.209.130,1 :
prevezmi hovor (102));
exten => 201,n,          JabberSend(asterisk,bob@192.168.209.130,2 :
posli na Alici 101);
exten => 201,n,          Set(OPTION=$
{JABBER_RECEIVE(asterisk,bob@192.168.209.130)});
exten => 201,n,          GotoIf(${[OPTION] = 2}?busy:10)
exten => 201,10,         JabberSend(asterisk,bob@192.168.209.130,
(Volam 102...));
exten => 201,n,          Dial(SIP/102,30);
exten => 201,n,          VoiceMail(6001@mailbox-102)
exten => 201,n,          Hangup();

;Bob neni dostupny v kalendari nebo na chatu, je dostupna Alice?
Pokud ano, volej ji
exten => 201,n(busy),Set(STATUS=$
{JABBER_STATUS(asterisk,alice@192.168.209.130)});
exten => 201,n,gotoif(${[${STATUS}] < 7}?105:200)
exten => 201,105,
JabberSend(asterisk,alice@192.168.209.130,Bob je nedostupny,
prichozi hovor od uzivatele ${CALLERID(name)},vyber cilovou
pobočku:);
exten => 201,106,         JabberSend(asterisk,alice@192.168.209.130,1
: prevezmi hovor (101));
exten => 201,107,         JabberSend(asterisk,alice@192.168.209.130,2
: posli na do hlasove schranky Boba);
exten => 201,108,         Set(OPTION=$
{JABBER_RECEIVE(asterisk,alice@192.168.209.130)});
exten => 201,109,         GotoIf(${[OPTION] = 2}?200:110)
exten => 201,110,         JabberSend(asterisk,alice@192.168.209.130,
(Volam 101...));
exten => 201,n,          Dial(SIP/101,30);
exten => 201,n,          VoiceMail(6001@mailbox-102)
exten => 201,n,          Hangup();

;Bob ani Alice nejsou dostupni
exten => 201,200,         VoiceMail(6001@mailbox-102)
exten => 201,n,          Hangup();

```

## 4 Závěr

Cílem diplomové práce bylo popsat pokročilé služby, které dnešní moderní telefonní ústředny mohou zajistit a implementace pokročilých služeb pro softwarovou ústřednu Asterisk. Výstupem je sada scénářů a odladěných nastavení, které zajišťují plnou funkcionalitu a interoperabilitu použitých komponentů s Asteriskem. Integrací hlasových služeb do již používaných informačních systémů lze docílit rychlejší a kvalitnější komunikace jak uvnitř společnosti tak i společnosti se svými klienty. Propojování telefonní ústředny s jinými komunikačními řešeními nám pomáhá lépe obsluhovat směrování příchozích hovorů pro daného uživatele na základě jeho aktuálního stavu.

Pro účely praktické části, jejíchž cílem byla prezence založená na XMPP a služby kalendáře, byla použita pobočková ústředna Asterisk. Oproti komerčním řešením za statisíce korun je tento projekt dostupný zdarma a vyvíjen jako opensource. Asterisk byl nainstalován na virtuálním stroji v mém PC v operačním systému Ubuntu stejně jako XMPP server Openfire a kalendářový server Davical. Tyto aplikace jsou také zdarma a mají otevřený kód. Na hostovaném PC běželi klienti využívající služeb serveru.

Po provedené instalaci a konfiguraci jednotlivých aplikací byly mezi sebou vzájemně propojeny. Tímto propojením byly realizovány následující praktické služby:

Automatická změna stavu XMPP klienta na On Phone, pokud má uživatel aktivní hovor na klapce Asterisku, kterou má spojenou s XMPP účtem. Tímto jsou ostatní spolupracovníci upozorněni, že daný uživatel momentálně není plně dostupný.

Přímé volání účastníka bez znalosti jeho telefonního čísla. V XMPP klientu Spark je možné, pokud má uživatel v systému přidělenou pobočku, přímo volat daného účastníka. Po zvolení tlačítka volat ústředna nejprve zavolá volajícímu a po vyzvednutí hovoru dochází k volání požadovaného účastníka.

Oznámení uživatele v okně XMPP klienta o právě probíhajících hovorech. Tímto způsobem je možné online kontrolovat provoz určitého účastníka. Stejně tak je možné v okně upozornit volajícího o nedostupnosti volaného.

Směrování hovoru na pobočku určenou odpovědí uživatele v okně XMPP klienta. Uživatel je v okně obeznámen, že má příchozí hovor a je mu nabídnuta možnost volby směrování na cílovou pobočku. Na základě jeho číselné odpovědi je hovor směrován na požadovanou pobočku.

Směrování hovoru na základě stavu nastaveného u XMPP klienta. Protokol XMPP je schopen nastavit až 7 různých stavů. Dle těchto stavů se mění chování Asterisku a hovory se směřují na různé linky.

Směrování hovoru na základě stavu nastaveného u události v online kalendáři. Asterisk pravidelně kontroluje dostupnost uživatele v kalendáři (stav volno nebo zaneprázdněnost) a podle toho směřuje hovor na různé pobočky.

Zápis detailů o hovoru do kalendáře. Po skončení hovoru je do zvoleného kalendáře automaticky zapsán čas počátku a konce hovoru a jména účastníků, kteří spolu komunikovali.

Automatické upozornění na schůzku. Asterisk nastavený čas před schůzkou zavolá uživateli a přečte mu detaily, které má zapsané u události v kalendáři.

Souhrn schůzek na následující hodinu. Po zavolání na pobočku 888 Asterisk přečte události, které má uživatel zapsány na následující hodinu.

Prezenční skript, který je souhrnem výše uvedených funkcí. Jedná se o modelovou situaci vedoucího, který má svou asistentku. Pokud přichází hovor na vedoucího, Asterisk zkontroluje jeho stav kalendáři a XMPP klientu. Pokud je dostupný, přijde mu zpráva o přichozím hovoru a může se rozhodnout, jestli hovor přijme. Pokud není jakákoliv podmínka splněna, hovor přechází na asistentku, zkontroluje se její stav v XMPP klientu, a hovor jde buď na ni nebo do hlasové schránky.

Projekt je možné po dopracování konfiguračního prostředí uživatelů integrovat do stávající pobočkové ústředny Asterisk v telefonní síti VŠB-TU Ostrava. Díky podrobné dokumentaci praktické části práce a dostupnému obrazu virtuálního stroje je možné snadno projekt rekonstruovat a navázat na něj v dalších projektech a pracích.

## 5 Použitá literatura

- [1] VOZŇÁK, M. *Voice over IP*. Vyd. 1. Ostrava : VŠB-TU Ostrava, 2008. 176 s. ISBN 978-80-248-1828-3.
- [2] MEGGELEN, J., SMITH, J., MADSEN, L. *Asterisk: The Future of Telephony*. O'Reilly Media, 2nd Edition, 2007. ISBN 978-05-965-1048-0.
- [3] GOMILLION, D.; DEMPSTER, B. *Building Telephony Systems with Asterisk : An easy introduction to using and configuring Asterisk to build feature-rich telephony systems for small and medium businesses*. Vyd. 2. Birmingham : Packt Publishing, 2005. 180 s. ISBN 1-904811-15-9.
- [4] VOZŇÁK, M. *Spojovací systémy*. Vysokoškolská skripta, 196 str. Vydavatel: VŠB-TU Ostrava, 1. vydání, únor 2009, ISBN 978-80-248-1961-7.
- [5] SHARIF, Ben. *Elastix without tears* [online]. Vyd. 1. Australia : Vl.n., 2008 [cit. 2011-04-24]. Dostupné z WWW: <[http://sourceforge.net/projects/elastic/files/Tutorials\\_Docs\\_Manuals/Elastix%20Without%20Tears/elastic\\_without\\_tears\\_june2010.pdf/download](http://sourceforge.net/projects/elastic/files/Tutorials_Docs_Manuals/Elastix%20Without%20Tears/elastic_without_tears_june2010.pdf/download)>.
- [6] SHARMA, Mayank. *Openfire Administration*. Vyd. 1. Birmingham : Packt Publishing Ltd, 2008. 233 s. ISBN 978-1-847195-26-5.
- [7] *Wiki.asterisk.org* [online]. 2010 [cit. 2011-03-24]. Asterisk 1.8 Documentation. Dostupné z WWW: <<https://wiki.asterisk.org/wiki/display/AST/Home>>.
- [8] JUSTRUMOURS, . *Voip-Info.org* [online]. c2010, 1.2.2010 [cit. 2011-04-24]. Asterisk TAPI. Dostupné z WWW: <<http://www.voip-info.org/wiki/view/Asterisk+TAPI>>.
- [9] *Wiki.ubuntu.cz* [online]. 2011 [cit. 2011-03-24]. PostgreSQL. Dostupné z WWW: <<http://wiki.ubuntu.cz/PostgreSQL>>.
- [10] *Wiki.davical.org* [online]. 2011 [cit. 2011-03-24]. Ubuntu/Lucid. Dostupné z WWW: <<http://wiki.davical.org/w/Ubuntu/Lucid>>.
- [11] *Www.ubuntugeek.com* [online]. 2010 [cit. 2011-03-24]. Howto install Sun Java Runtime Environment (JRE) in Ubuntu 10.04 (Lucid Lynx). Dostupné z WWW: <<http://wiki.davical.org/w/Ubuntu/Lucid>>.

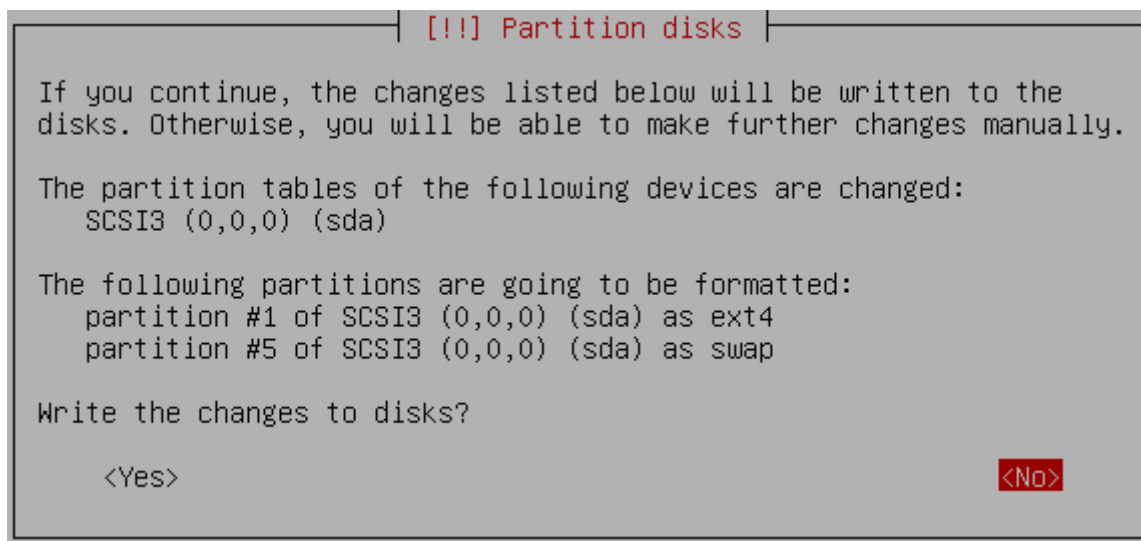


- [12] *KIKLHORN, M. Forum.ubuntu.cz* [online]. 2010 [cit. 2011-03-24]. Instalace Festival-czech .Dostupné z WWW: <<http://forum.ubuntu.cz/index.php?action=printpage;topic=49568.0>>.
- [13] *Www.asterisk2billing.org* [online]. 2011 [cit. 2011-03-24]. A2Billing. Dostupné z WWW: <<http://www.asterisk2billing.org>>.
- [14] *Www.vmware.com* [online]. 2011 [cit. 2011-03-24]. Vmware workstation. Dostupné z WWW: <<http://http://www.vmware.com/products/workstation/>>.
- [15] *Www.mozilla.org* [online]. 2011 [cit. 2011-03-24]. Mozilla Sunbird . Dostupné z WWW: <<http://www.mozilla.org/projects/calendar/sunbird/>>.
- [16] ČERVENKA, Marek. *Abclinuxu.cz* [online]. 15.12.2009 [cit. 2011-03-24]. Asterisk a kalendáře. Dostupné z WWW: <<http://www.abclinuxu.cz/clanky/navody/asterisk-a-kalendare>>.

## 6 Seznam příloh

**Všechny přílohy jsou dostupné na DVD ve složce přílohy**

**Příloha č.1:** Rozdělení disku při instalaci operačního systému



## Příloha č.2: Konfigurace XMPP klienta Pidgin

**Přidat účet**

Základní Pokročilé Proxy

**Možnosti přihlášení**

Protokol: XMPP

Jméno uživatele: alice

Doména: 192.168.209.130

Zdroj:

Heslo: •••••


☐ Zapamatovat si heslo

**Nastavení uživatele**

Místní alias:

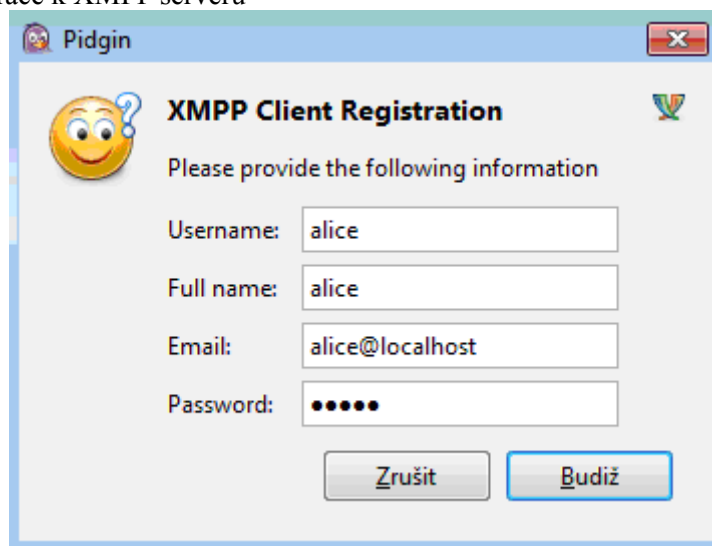
☐ Upozornění na nové zprávy

☐ Používat pro tento účet tuto ikonu kamaráda:



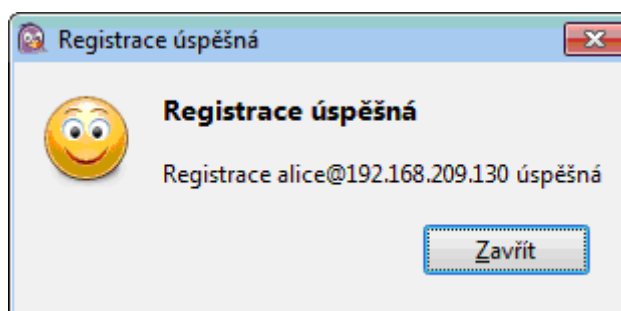
☒ Vytvořit tento nový účet na serveru

**Příloha č. 3:** Registrace k XMPP serveru



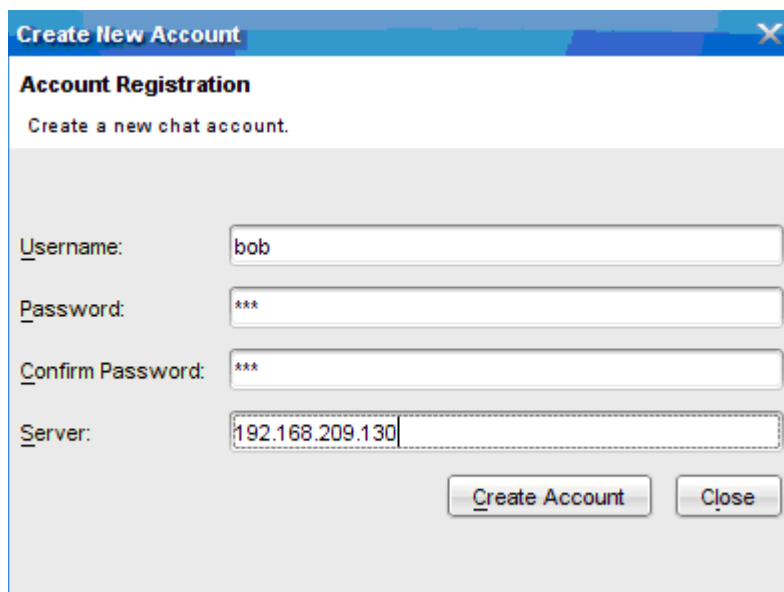
The image shows a dialog box titled "Pidgin" with a subtitle "XMPP Client Registration". It contains a question mark icon and the text "Please provide the following information". There are four input fields: "Username:" with the value "alice", "Full name:" with the value "alice", "Email:" with the value "alice@localhost", and "Password:" with masked characters "•••••". At the bottom, there are two buttons: "Zrušit" (Cancel) and "Budiž" (OK).

**Příloha č.4:** Potvrzení registrace k XMPP serveru



The image shows a dialog box titled "Registrace úspěšná" (Registration successful). It contains a smiley face icon and the text "Registrace úspěšná" and "Registrace alice@192.168.209.130 úspěšná". At the bottom, there is a button labeled "Zavřít" (Close).

**Příloha č.5:** Konfigurace XMPP klienta Spark



The image shows a dialog box titled "Create New Account" with a subtitle "Account Registration". It contains the text "Create a new chat account." and four input fields: "Username:" with the value "bob", "Password:" with masked characters "\*\*\*", "Confirm Password:" with masked characters "\*\*\*", and "Server:" with the value "192.168.209.130". At the bottom, there are two buttons: "Create Account" and "Close".

## Příloha č.6: Konfigurace SIP klients X-lite

The image shows a screenshot of the 'Properties of Account1' dialog box in the X-Lite application. The dialog has a title bar with a close button (X). Below the title bar are five tabs: 'Account', 'Voicemail', 'Topology', 'Presence', and 'Advanced'. The 'Account' tab is selected and active.

Under the 'Account' tab, there are three main sections:

- User Details:** This section contains five text input fields:
  - Display Name:** Contains the text 'alice'.
  - User name:** Contains the text '101'.
  - Password:** Contains five dots (•••••).
  - Authorization user name:** Contains the text '101'.
  - Domain:** Contains the text '192.168.209.130'.
- Domain Proxy:** This section contains a checked checkbox labeled 'Register with domain and receive incoming calls'. Below this is the text 'Send outbound via:' followed by three radio button options:
  - domain:** Unselected.
  - proxy:** Unselected. To its right is an empty text input field labeled 'Address'.
  - target domain:** Selected (indicated by a filled circle).
- Dialing plan:** This section contains a single text input field with the value '#1\|a\|a.T;match=1;prestrip=2;'.

At the bottom of the dialog, there are three buttons: 'OK', 'Storno', and 'Použít'.

**Příloha č.7:** Kontrola úspěšné registrace SIP účtu



## Příloha č. 8: Konfigurace Express Talk klienta

The screenshot shows the 'Options' dialog box with the 'Lines' tab selected. The 'Lines' tab is highlighted in the top navigation bar. The 'Line Settings' section contains instructions and a dropdown menu for 'Settings for Line' set to 'Default Line Settings'. A checked checkbox 'Use the Default Line Settings for this line' is present. Below are text input fields for 'Full Friendly Display Name' (Bob), 'Server (SIP Proxy or Virtual PBX)' (192.168.209.130), 'SIP Number (or User Name)' (102), and 'Password' (masked with dots). A button labeled 'Advanced Line Settings' is below the password field. At the bottom, there is an unchecked checkbox 'Check for voice mail notifications on the default line (Requires Axon and IVN)' and a paragraph of advice with two links: 'Open recommended Free SIP Servers Web Page (for internet calls)' and 'Download Virtual PBX Software (usually for Offices or Call Centers)'. The bottom of the dialog has 'OK', 'Storno', and 'Nápověda' buttons.

**Options**

Emergency Other Speed Dial CRM

Audio Devices Video **Lines** Network Hold/Record Control

**Line Settings**

By default all lines use the Default Line Settings.  
To change default settings, select Default Line Settings in the list.

Settings for Line: Default Line Settings

☒ Use the Default Line Settings for this line

Full 'Friendly' Display Name: Bob  
eg. Jane Doe, ACME Inc

Server (SIP Proxy or Virtual PBX): 192.168.209.130  
eg. computer.network, proxy.domain.com

SIP Number (or User Name): 102  
eg. 555123456 (or Jane.Doe)

Password: ●●●

Advanced Line Settings

☐ Check for voice mail notifications on the default line (Requires Axon and IVN)

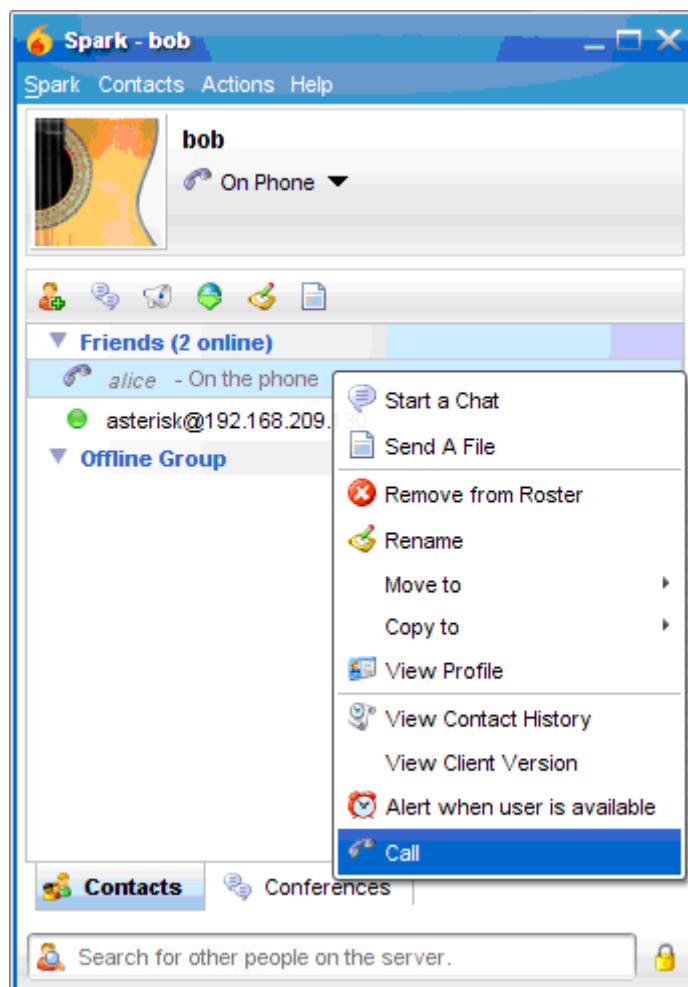
If you are in doubt about any of the above settings, please see the email from the SIP company you signed up with (or see Help).

[Open recommended Free SIP Servers Web Page \(for internet calls\)](#)

[Download Virtual PBX Software \(usually for Offices or Call Centers\)](#)

OK Storno Nápověda

**Příloha č.9:** Klient Spark – stav „On Phone“ a přímé volání



**Příloha č.10:** Přehled použitých příkazů - soubor příkazy.txt na dvd ve složce Texty



# Adresářová struktura přiloženého DVD

/ Přílohy – přílohy 1 – 9

/ Software – instalační soubory

- pidgin-2.7.10.exe
- spark\_2\_5\_8.exe
- Sunbird Setup 1.0 Beta 1.exe
- X-Lite\_Win32\_1003l\_30942.exe

/ Texty – diplomová práce a přehled příkazů

- Diplomová práce.pdf
- Příkazy.txt

/ Virtuální server – obraz virtuálního serveru spustitelný v programu VMWare